arXiv:2602.07697v3 [cs.LG] 8 Aug 2026

On the Infinite Width and Depth Limits of Predictive Coding Networks

Francesco Innocenti [1 2] **El Mehdi Achour** [3] **Rafal Bogacz** [1 2]

Abstract

Predictive coding (PC) is a biologically plausible alternative to standard backpropagation (BP) that minimises an energy function with respect to network activities before updating weights. Recent work has improved the training stability of deep PC networks (PCNs) by leveraging some BP-inspired reparameterisations, but the scalability and theoretical basis of these methods remain unclear. To address this gap, we study the infinite width and depth limits of PCNs. For linear networks, we derive stable and "non-lazy" parameterisations when scaling both the model width and depth, revealing that the output of standard PCNs explodes with width during training. Moreover, under stable parameterisations, we show that the gradients computed by PC at activity equilibrium converge to the BP gradients for networks that are much wider than deep ($depth/width \to 0$). Experiments show high gradient alignment between PC and BP at large width for different nonlinear models, including convolutional networks and transformers. Overall, this work constrains the parameterisations that are scalable with PC, while suggesting how BP could be implemented using only local updates in much wider than deep networks like the brain.

1. Introduction

How does the brain update synaptic weights using only local information available to each neuron? While backpropagation (BP) remains the standard algorithm for training artificial neural networks (Rumelhart et al., 1986; LeCun et al., 2015), it is unlikely to be implemented in the brain due to its non-local propagation of information (Lillicrap et al., 2020). Understanding how the brain solves this credit assignment problem would not only greatly advance neuroscience, but could also unlock more energy efficient AI.

A more biologically plausible alternative to BP is "predictive coding" (PC), an influential theory of brain function suggesting that neurons minimise their prediction errors (Rao & Ballard, 1999; Friston, 2005). In a predictive coding network (PCN), this process occurs in two phases: first, neurons adjust their activity to minimise a sum of local objectives (or energies); then, once the network has reached an equilibrium, weights are updated to minimise the same energy function (Song et al., 2024). An advantage of PC and other local algorithms is that, at equilibrium, the weights of different layers can be updated in parallel.

While the conditions under which PC can approximate or exactly equal BP are now well-established (Whittington & Bogacz, 2017; Song et al., 2020; Millidge et al., 2022b; Rosenbaum, 2022; Salvatori et al., 2021; Millidge et al., 2022a), later work explored how standard PC could provide benefits over BP, including learning in fewer weight updates (Alonso et al., 2022; Innocenti et al., 2023; Alonso et al., 2023; Innocenti et al., 2024a) and increased performance in online and continual learning tasks (Song et al., 2024).

Encouraged by these potential advantages, recent efforts have focused on scaling PC to larger and especially deeper models (Pinchetti et al., 2024). This work has revealed, and at least partially addressed, various instabilities related to the training of deep PCNs (Qi et al., 2025; Innocenti et al., 2025; Goemaere et al., 2025). Notably, Innocenti et al. (2025) proposed a BP-derived reparameterisation of PCNs allowing stable training of 100+ layer residual networks on simple classification tasks. However, this and other reparameterisations tend to be heuristically derived, or only partially justified, and the scalability of PC to larger datasets (e.g. ImageNet) still remains to be seen.

The successful scaling of BP-trained models has been, in no small part, driven by parameterisations derived in "idealised" infinite width and depth limits (Yang & Hu, 2021; Bordelon & Pehlevan, 2022b; Bordelon et al., 2023; Dey et al., 2025). These parameterisations enable not only stable training dynamics across model sizes, but also "zero-shot hyperparameter transfer" (Yang et al., 2021), whereby optimal tuning parameters such as the learning rate remain approximately constant across scales.

[1]Brain Network Dynamics Unit, University of Oxford, UK [2]MRC CoRE in Restorative Neural Dynamics, UK [3]UM6P College of Computing, Rabat, Morocco. Correspondence to: Francesco Innocenti <francesco.innocenti@ndcn.ox.ac.uk>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

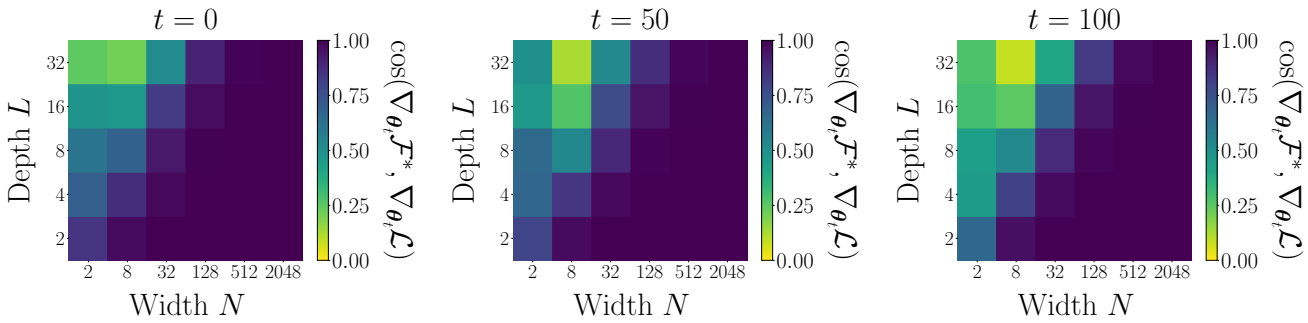


Figure 1. **PC shows perfect gradient alignment with BP for vanishing depth-to-width ratios ($L/N \to 0$), under stable and feature-learning parameterisations of linear residual networks.** We trained linear residual networks on CIFAR-10 with the mean-field parameterisation (as defined in Table 2) and depth scaling exponent $\alpha = 1/2$ (§4). Plotted are the cosine similarities averaged over 3 random runs between the equilibrated energy (Eq. 5) gradients (PC) and the MSE loss (Eq. 1) gradients (BP) at different training steps t. See §A.9 for more details and Figure A.6 for similar results on Fashion-MNIST.

Moreover, analyses of infinitely wide networks have already provided insights into the learning dynamics of different bio-plausible learning rules (Bordelon & Pehlevan, 2022a) including PC (Ishikawa et al., 2024). Given this, and the practical impact of theoretical parameterisations for BP (Yang et al., 2021), this leads us to the following question:

Can we derive principled parameterisations specific to PC that allow stable, non-trivial and potentially more efficient training of both wide and deep networks?

Here, we positively answer this question by studying the infinite width N and depth L limits of PCNs. For linear networks, we theoretically derive stable and feature-learning ("non-lazy") parameterisations for PC when scaling both the width and depth. Surprisingly, we find that the same set of parameterisations as for BP ensures stability and non-trivial feature learning (as defined in §2.3 & §4) for PC. This has the important practical implication that the predictions of networks trained with PC using the standard (e.g. PyTorch) parameterisation blow up with the width, just as for BP.

In addition, under stable parameterisations, we show that the gradients computed by PC converge to the BP gradients for much wider than deep residual networks, with an $O(L/N)$ relative error. We demonstrate that our theoretical results hold in practice for nonlinear models, including convolutional neural networks (CNNs) and transformers, trained with different nonlinearities, optimisers and loss functions. Overall, our work constrains the parameterisations that are scalable with PC, while suggesting a way in which BP could be implemented using a local algorithm in much wider than deep networks like the brain.

The rest of the paper is organised as follows. After some background on PCNs and parameterisations for BP, Section 3 presents our results on the width parameterisation of linear multi-layer perceptrons (MLPs) for PC. This is followed by results on the depth PC parameterisation of linear residual networks (§4). We then present experiments with nonlinear models supporting our theoretical results (§5), before concluding with the implications and limitations of this work (§6). For space reasons, we refer related work, additional experiments and derivations to Appendix A.

1.1. Summary of contributions

- For linear MLPs with equilibrated activities, we prove that the set of stable and feature-learning parameterisations for PC when scaling the model width is the same as for BP (Theorem 1). Moreover, under any such parameterisation, we show that the gradients computed by PC converge to the BP gradients as the width grows to infinity, $N \to \infty$ (Corollary 3.2; Figures 2 & A.15).
- We generalise these results to linear residual networks, proving that the set of stable and feature-learning parameterisations for PC when scaling *both* the width and depth is also the same as for BP (Theorem 2; see also Figure A.5). Similar to the MLP case, we show that the PC gradients converge to BP's for networks with a vanishing depth-to-width ratio, $L/N \to 0$ (Corollary 4.2; Figures 1 & A.6).
- We empirically validate our theoretical results on linear and nonlinear architectures including CNNs and transformers (e.g. Figure 4), trained on toy and large-scale tasks (e.g. ImageNet), with different optimisers and loss functions. We provide heuristic extensions of our derived PC parameterisations to CNNs, transformers, and the Adam optimiser (§A.7).

2. Background

To set up our theoretical analysis of PCNs (§3), in this section we review PC, define the general parameterisation we consider, and revisit previous "width-aware" parameterisations for BP. For general notation, see §A.1.

2.1. Predictive coding networks

We consider the standard supervised learning setting given a dataset of P examples $\mathcal{D} = \{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^P$, with $\boldsymbol{x}_\mu \in \mathbb{R}^D$ and scalar targets $y_\mu \in \mathbb{R}$, which we assume for simplicity throughout. For comparison, it is useful to recall the standard mean squared error (MSE) loss

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2P} \sum_{\mu=1}^{P} (y_\mu - f(\boldsymbol{x}_\mu; \boldsymbol{\theta}))^2, \tag{1}$$

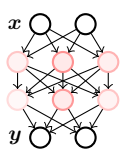
where $f(\boldsymbol{x}_\mu; \boldsymbol{\theta})$ denotes a model's prediction for a given input, with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$.

PC energy. PCNs minimise a sum of local objectives (energies) that typically take the form of layer-wise MSEs (Buckley et al., 2017; Bogacz, 2017). For concreteness, consider the PC energy of a simple MLP with no biases

$$\mathcal{F}(\boldsymbol{z}, \boldsymbol{\theta}) = \frac{1}{2P} \sum_{\mu=1}^{P} \sum_{\ell=1}^{L} \left\| \boldsymbol{z}_\mu^{(\ell)} - \phi_\ell(\boldsymbol{W}^{(\ell)} \boldsymbol{z}_\mu^{(\ell-1)}) \right\|_2^2, \tag{2}$$

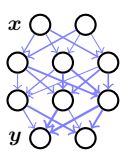
where $\boldsymbol{z}_\mu := \{\boldsymbol{z}_\mu^{(\ell)}\}_{\ell=0}^L$ denote the activities of neurons that are potentially free to vary, $\{\boldsymbol{W}^{(\ell)}\}_{\ell=1}^L$ are weight matrices, and $\phi_\ell(\cdot)$ is an element-wise layer activation function.

PC inference & learning. For supervised learning, PCNs are trained by clamping the first and last layer to some input and target data: $z_\mu^{(L)} \leftarrow y_\mu$ and $\boldsymbol{z}_\mu^{(0)} \leftarrow \boldsymbol{x}_\mu$. The energy (Eq. 2) is then minimised in a bi-level, expectation-maximisation fashion. First, we minimise the energy with respect to the activities of the network,



$$\boldsymbol{z}_\mu^* = \arg\min_{\boldsymbol{z}_\mu} \mathcal{F}(\boldsymbol{z}_\mu, \boldsymbol{\theta}_t). \tag{3}$$

This process is called "inference" and can be intuitively thought of as the network trying to find an equilibrium of its activity that best accounts for each data sample. Eq. 3 is typically minimised using standard gradient descent (GD): $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \beta \nabla_{\boldsymbol{z}} \mathcal{F}$ with some step size β, where for simplicity we drop the data index μ. After convergence of the activities $\boldsymbol{z}_\mu^*$, we minimise the energy with respect to the weights, by performing a single (e.g. GD) update:



$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\partial \mathcal{F}(\boldsymbol{z}^*, \boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}, \tag{4}$$

where η is some learning rate. The optimisation cycle is then restarted with a new data batch. Depending on the task (e.g. discrimination vs generation), the network can be tested to predict a target given some input, or to infer an input given some target. Note that, in contrast to BP, the gradients of the energy with respect to both the activities and weights of any layer are *local*, in the sense that they only require information from adjacent layers.

Equilibrated energy for linear networks. For arbitrary linear networks with $\phi_\ell(\cdot) = \boldsymbol{I}$ for all ℓ, the inference minimisation problem of Eq. 3 has a unique solution $\boldsymbol{z}_\mu^*(\boldsymbol{\theta})$ where $\partial \mathcal{F} / \partial \boldsymbol{z}_\mu^* = \boldsymbol{0}$ (Ishikawa et al., 2024; Innocenti et al., 2025). Moreover, as shown by Innocenti et al. (2024a), the PC energy (Eq. 2) evaluated at this solution $\mathcal{F}(\boldsymbol{z}_\mu^*(\boldsymbol{\theta}), \boldsymbol{\theta})$ is equal to a rescaled MSE loss with a non-trivial, weight-dependent rescaling, as shown below. For a linear MLP with scalar output, this *equilibrated energy*, which abbreviate as $\mathcal{F}^*(\boldsymbol{\theta})$, takes the following form:

$$\mathcal{F}^*(\boldsymbol{\theta}) = \frac{\mathcal{L}(\boldsymbol{\theta})}{s(\boldsymbol{\theta})}, \tag{5}$$

$$s(\boldsymbol{\theta}) = 1 + \sum_{\ell=2}^{L} \|\boldsymbol{W}^{(L:\ell)}\|_2^2, \tag{6}$$

where we define the product $\boldsymbol{W}^{(L:\ell)} := \boldsymbol{w}^{(L)} \boldsymbol{W}^{(L-1)} \cdots \boldsymbol{W}^{(\ell)}$ for $\ell \in \{1, \dots, L\}$. For intuition, the rescaling for a one-hidden-layer network simplifies to the squared norm of the output weights plus one, $s(\boldsymbol{\theta}) = 1 + \|\boldsymbol{w}^{(L)}\|^2$. Importantly, Eq. 5 is the objective on which PC effectively learns for linear networks, and insights obtained from the equilibrated energy have been empirically shown to transfer to nonlinear PCNs (Innocenti et al., 2024a). For this reason, Eq. 5 will be the basis of all our theoretical results.

2.2. General parameterisation

We begin by studying linear MLPs of width N and depth L with scalar output. We will empirically show that our theory holds for multidimensional outputs and different nonlinear networks. Our setting and notation are closest to those of Bordelon & Pehlevan (2022b; 2025). In particular, we consider the following general MLP parameterisation:

$$f_\mu = \frac{1}{\gamma} h_\mu^{(L)}, \tag{7}$$

$$h_\mu^{(L)} = \frac{1}{N^{a_L}} \boldsymbol{w}^{(L)} \boldsymbol{h}_\mu^{(L-1)}, \tag{8}$$

$$\boldsymbol{h}_\mu^{(\ell)} = \frac{1}{N^{a_\ell}} \boldsymbol{W}^{(\ell)} \boldsymbol{h}_\mu^{(\ell-1)}, \tag{9}$$

$$\boldsymbol{h}_\mu^{(1)} = \frac{1}{N^{a_1} \sqrt{D}} \boldsymbol{W}^{(1)} \boldsymbol{x}_\mu, \tag{10}$$

Table 1. Width-aware scaling exponents for MLPs of width N

Exponent	Scaling term	Description
a_ℓ	Preactivation	Determines the N^{-a_ℓ} scaling of the layer preactivations.
b_ℓ	Init. variance	Determines the N^{-b_ℓ} scaling of the weight init. variance.
c	Learning rate	Part of $\eta = \eta_0 \gamma^2 N^{-c}$, scaling the global learning rate.
d	Network output	Part of $\gamma = \gamma_0 N^d$, scaling the network predictions.

with layer activations[1] $\boldsymbol{h}_\mu^{(\ell)} \in \mathbb{R}^N$ for $\ell \in \{1, \dots, L-1\}$, and weight matrices $\boldsymbol{W}^{(1)} \in \mathbb{R}^{N \times D}$, $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{N \times N}$ and $\boldsymbol{w}^{(L)} \in \mathbb{R}^{1 \times N}$. Biases are ignored for simplicity. The exponents $a_\ell \in \mathbb{R}$ determine how the activations are scaled with the network width N. As we will see, the output scaling

$$\gamma = \gamma_0 N^d \tag{11}$$

determines the learning regime (rich vs lazy), with γ_0 as a width-independent constant controlling the strength or "richness" of feature learning, and d as a width scaling exponent. All the weights are initialised i.i.d. as

$$W_{ij}^{(\ell)} \sim \mathcal{N}(0, N^{-b_\ell}), \quad \ell \in \{1, \dots, L\}, \tag{12}$$

where b_ℓ determines how the variance is scaled with the width. Note that the "standard parameterisation" (SP) used in practice (e.g. in PyTorch) assumes $b_\ell = 1$ for $\ell > 1$ and $a_\ell = 0$ for all ℓ, since standard initialisations (LeCun et al., 2002; He et al., 2015) ensure that weights have variance inversely proportional to their input dimension. Lastly, we will in the first instance consider gradient flow (i.e. continuous-time GD)

$$\frac{d\boldsymbol{\theta}}{dt} = -\eta \frac{\partial l(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{13}$$

where depending on the context, $l(\boldsymbol{\theta})$ will denote the MSE loss (Eq. 1) or the equilibrated energy (Eq. 5). Our results are straightforward to generalise to other optimisers, and we provide a simple extension for Adam (Kingma & Ba, 2014) in §A.7. Following Bordelon & Pehlevan (2022b), the learning rate is parameterised as

$$\eta = \eta_0 \gamma^2 N^{-c}, \tag{14}$$

where η_0 is a constant, the γ^2 factor is included to make the change in the loss at initialisation independent of γ (see §A.4.3), and c is another width scaling exponent.

A (width-aware) parameterisation is simply defined by the set of scaling exponents (a_ℓ, b_ℓ, c, d), as summarised in Table 1. We will extend this to depth in §4. Following the analyses for BP (reviewed in the next section), our goal will be to derive parameterisations, or solve for the exponents, for PC under specific constraints or desiderata.

[1]Note that for linear networks there is no distinction between pre- and post-activations.

2.3. Width-aware parameterisations for BP

Having defined our general "abcd" parameterisation (§2.2), we now review the influential notion of a "**stable and feature-learning parameterisation**", also known as the "maximal update" (μP) (Yang & Hu, 2021) or "mean-field" (Bordelon & Pehlevan, 2022b) parameterisation. This type of parameterisation is defined by satisfying 3 main desiderata that are aimed at ensuring both stable and non-trivial learning dynamics in the infinite-width limit (Yang & Hu, 2020; 2021; Bordelon & Pehlevan, 2022b).

Parameterisation desiderata at large width N.

- **Desideratum 1.** Layer preactivations are stable with respect to the width at initialisation: $h_i^{(\ell)} = \Theta_N(1)$.
- **Desideratum 2.** Network predictions are width-stable during training: $df/dt = \Theta_N(1)$.
- **Desideratum 3.** Layer features are also width-stable during training: $dh_i^{(\ell)}/dt = \Theta_N(1)$.

Here $h_i^{(\ell)}$ refers to the ith neuron in the ℓth layer, and the notation $x = \Theta_N(1)$ indicates that x neither vanishes nor explodes (it is hence "stable") as $N \to \infty$ (see §A.1 for a definition).[2] The first two constraints relate to the numerical stability of the model with the width, while the last desideratum characterises the degree to which the features at each layer evolve during training. Parameterisations that violate (1) or (2) are therefore commonly said to be "unstable", while those that do not learn features are labelled as trivial or "lazy". For BP, there is a known set of one-dimensional parameterisations that satisfy all 3 desiderata (Yang & Hu, 2021; Bordelon & Pehlevan, 2022b):

$$\begin{cases} 2a_\ell + b_\ell = 1 & \text{for} \quad \ell \in \{2, \dots, L\}, \quad 2a_1 + b_1 = 0, \\ 2a_\ell + c = 1 & \text{for} \quad \ell \in \{2, \dots, L\}, \quad 2a_1 + c = 0, \\ d = 1/2. \end{cases} \tag{15}$$

That is, if one fixes any among (a_ℓ, b_ℓ, c), then there is a unique solution. For example, if one chooses $a_\ell = 1/2$ for

[2]This is similar to requiring that the average squared (Euclidean) norm of each layer preactivation is $\frac{1}{N}\|\boldsymbol{h}^{(\ell)}\|^2 = \Theta_N(1)$.

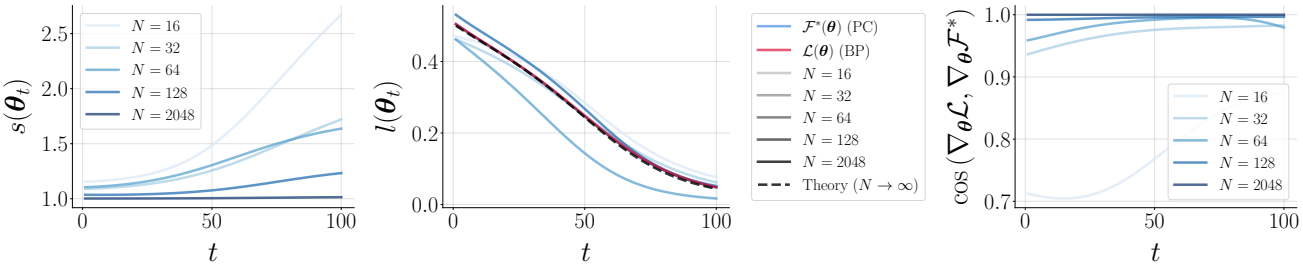


Figure 2. **PC converges to BP at large width, under width-stable and feature-learning parameterisations of linear MLPs.** We trained deep linear MLPs ($L = 5$) of varying widths N with full-batch GD on a toy task with binary labels. All models used the mean-field parameterisation as defined in Table 2. For comparative results with the SP, see Figure A.19. (*Left*) As predicted by Eq. 17, the equilibrated energy rescaling $s(\boldsymbol{\theta})$ approaches one as $N \to \infty$. (*Middle*) As a result, the equilibrated energy $\mathcal{F}^*(\boldsymbol{\theta})$ (Eq. 5) converges to the MSE loss $\mathcal{L}(\boldsymbol{\theta})$ (Eq. 1), and PC effectively computes the same gradients as BP (*Right*). The theoretical loss was calculated using dynamical mean field theory (see §A.9 for more details). For additional results including on CIFAR-10, see Figures A.18 & A.15.

$\ell > 1$, then one obtains the mean-field parameterisation considered by Bordelon & Pehlevan (2022b), with $b_\ell = c = a_1 = 0$. μP as introduced in Yang & Hu (2021) further requires that the learning rate η is $\Theta_N(1)$, leading to slightly different scalings. Note that the feature learning desideratum (3) is uniquely satisfied by $d = 1/2$, and this is the sense in which γ (Eq. 11) is said to determine the learning regime.

For comparison, note that the SP has stable initialisation as well as evolving features, but the network predictions explode with the width, as shown by Yang & Hu (2020). On the other hand, the so-called neural tangent kernel (NTK) parameterisation (Jacot et al., 2018) is fully stable (in that it satisfies the first 2 constraints), but its feature updates vanish with the width (hence known as lazy). For the specific scalings defining all these parameterisations, see Table 2.

3. Width-aware Parameterisations for PC

Given the empirical success of the width-stable and feature-learning BP parameterisations, we might want to ensure the same desiderata in §2.3 for PC. It turns out that, at least for linear networks under our considered parameterisations, PC admits exactly the same set of scalings as BP.

Theorem 1 (Width-stable and feature-learning parameterisations for linear PCNs.)**.** *Consider the (a_ℓ, b_ℓ, c, d) parameterisation of linear MLPs (Eqs. 7-10), and assume PCNs with converged activities that therefore learn on the equilibrated energy (Eq. 5). Then, there exists a set of one-dimensional parameterisations that satisfies the 3 Desiderata in §2.3 for PC, and this is the same as for BP (Eq. 15).*

See Figure A.5 for a visual illustration of the result. The full derivation is provided in §A.4 and involves checking every desideratum for linear PCNs performing gradient flow on the equilibrated energy (Eq. 5). Verification of the first desideratum does not differ between BP and PC (as the forward pass is algorithm-independent), while the other constraints involve taking into account extra terms arising from the rescaling of the equilibrated energy (Eq. 6).

As an intuition for Theorem 1, one can show that, as for BP, there is a unique output scaling exponent $d = 1/2$ that ensures non-lazy feature learning during training of PCNs. As we will see next, this scaling causes the equilibrated energy terms to become subleading compared to the loss terms, vanishing at a rate of $O(N^{-1})$.

3.1. PC convergence to BP for linear MLPs as $N \to \infty$

We just showed that the set of width-stable and feature-learning parameterisations for PC is the same as for BP (Theorem 1). Under any of these parameterisations, it is straightforward to show that the weight gradients computed by PC converge to BP's in the infinite-width limit. Here we take $N \to \infty$ at fixed depth L because, unlike residual networks, MLPs do not generally have depth-stable forward and backward passes, as discussed in §A.5.1.

Corollary 3.2 (PC convergence to BP on wide linear MLPs.)**.** *Under any width-stable and feature-learning parameterisation of a linear MLP (Eq. 15), at fixed depth the gradient of the equilibrated energy (Eq. 5) converges to the BP gradient of the MSE loss (Eq. 1) as $N \to \infty$. More precisely,*

$$\frac{\|\nabla_{\boldsymbol{\theta}}\mathcal{F}^* - \nabla_{\boldsymbol{\theta}}\mathcal{L}\|_2}{\|\nabla_{\boldsymbol{\theta}}\mathcal{L}\|_2} = O\left(N^{-1}\right), \qquad (16)$$

provided the MSE gradient remains non-vanishing.

This result is derived in §A.6. For intuition, consider the equilibrated energy rescaling (Eq. 6) for a one-hidden-layer PCN under the mean-field parameterisation (see Ta-

ble 2): $s(\boldsymbol{\theta}) = 1 + \gamma_0^{-2} N^{-2} \|\boldsymbol{w}^{(L)}\|^2$. Given the first desideratum's constraint (§2.3), the norm term scales as $\|\boldsymbol{w}^{(L)}\|^2 = \Theta(N)$, leading to

$$s(\boldsymbol{\theta}) = 1 + \frac{1}{\gamma_0^2 N} \Theta_N(1) = 1 + \Theta(N^{-1}) \tag{17}$$

$$\implies \quad \mathcal{F}^*(\boldsymbol{\theta}) \to \mathcal{L}(\boldsymbol{\theta}), \quad \text{as } N \to \infty. \tag{18}$$

We see that the energy rescaling approaches one in the infinite-width limit, which implies that the equilibrated energy (Eq. 5) converges to the MSE loss (Eq. 1). §A.6 shows that, at fixed depth, $O(1/N)$ convergence also holds for the gradients of the equilibrated energy, under stable and rich parameterisations. All these results are empirically verified in Figures 2, A.15 & A.21 for both linear and nonlinear networks trained on simple tasks. We note that a closely related result was proved by Ishikawa et al. (2024), which we discuss in more detail in §6 and §A.2.

> ***Takeaway 1****: The set of width-stable and feature-learning parameterisations for linear PCNs with equilibrated activities is exactly the same as for BP. Moreover, under any of these parameterisations, PC converges to BP at large width.*

3.2. Learning regimes

BP-like regimes. Since PC converges to BP under any width-stable feature-learning parameterisation as $N \to \infty$ (Corollary 3.2), it is perhaps not surprising that PC exhibits similar learning regimes. The output constant γ_0 (Eq. 11) can be used to interpolate between these different regimes:

- $\gamma_0 < 1$ induces kernel-like (lazy) behaviour as the rescaling $s(\boldsymbol{\theta})$ dominates the equilibrated energy (Eq. 5) compared to the MSE loss (Eq. 1), leading to zero energy in the limit of $\gamma_0 \to 0$.
- $\gamma_0 = 1$ is equivalent to μP up to one (a_ℓ, b_ℓ, c) degree of freedom and closely related to the SGD parameterisation for PC studied by Ishikawa et al. (2024).
- $\gamma_0 > 1$ leads to richer feature learning as $s(\boldsymbol{\theta})$ converges faster to 1, recovering the known small initialisation or "saddle-to-saddle regime" for BP (Saxe et al., 2013; Jacot et al., 2021).

Figure A.3 verifies these regimes for PC, showing that larger values of γ_0 are associated with faster learning. Next, we show how our general parameterisation can provide insights into a previously identified regime that is unique to PC.

Fast (but unstable) saddle-to-saddle PC regime. Recently, Innocenti et al. (2024a) showed that many degenerate saddle points of the MSE loss (Eq. 1) become benign in the equilibrated energy (Eq. 5), and so much easier to escape. For deeper than wide models ($L > N$), this leads to significantly faster saddle-to-saddle dynamics for PC than BP for small initialisation (and SGD with small learning rate) (Innocenti et al., 2024a). Such a regime can be recovered by noting that deeper than wide networks under the SP, as assumed by Innocenti et al. (2024a), are effectively initialised near the origin (see Figure A.4).

However, this fast PC saddle regime is unstable in two distinct ways. First, the more one would like to increase N, the more L would have to grow to preserve the PC speed-ups, as shown in Figure A.4. However, because this regime relies on the SP, the model is unstable at large width, in the sense of violating Desideratum 2 (§2.3). Moreover, this regime also grows unstable with the depth, in that the variance of the MLP preactivations is prone to vanish/explode except under highly restrictive conditions (Poole et al., 2016; Schoenholz et al., 2016) (see also Figures A.7 & A.13).

As we will see next, residual models can make the forward pass much more stable with the depth. However, as noted by Innocenti et al. (2024a), residual networks do not exhibit the same saddle regime as MLPs, since they effectively shift the position of the origin saddle (Hardt & Ma, 2016). We therefore reach the following conclusion.

> ***Takeaway 2****: The only regime where PC has been shown to provide potential benefits over BP in the form of "fast saddle-to-saddle dynamics" grows unstable with both width and depth.*

4. Depth-aware Parameterisations for PC

Following previous work for BP (Yang et al., 2023; Bordelon et al., 2023), we now extend our width PC parameterisations for MLPs (§3) to infinitely deep residual networks, which have a preactivation variance that can be more easily controlled with the depth. In what follows, we consider joint width-depth limits $N, L \to \infty$ satisfying $L/N \to 0$, where the PC-specific contribution to each direct feature update is asymptotically subleading (see §A.5.1). For simplicity, let us fix a specific width-stable feature-learning parameterisation with $a_\ell = 1/2$ for $\ell > 1$ (see Table 2). To extend our previous MLP parameterisation (§2.2), we simply redefine the layer activations for $\ell \in \{2, \dots, L-1\}$ (Eq. 9) as

$$\boldsymbol{h}_\mu^{(\ell)} = \left(\boldsymbol{I} + \frac{1}{L^\alpha \sqrt{N}} \boldsymbol{W}^{(\ell)} \right) \boldsymbol{h}_\mu^{(\ell-1)}, \tag{19}$$

where we introduced a (one-block) skip connection and a depth-dependent scaling exponent α. In such a parameterisation, it is typical to require the following two depth-specific desiderata (Yang et al., 2023; Bordelon et al., 2023).

Parameterisation desiderata at large depth L.

- **Desideratum 4.** Hidden activations are depth-stable at initialisation: $h_i^{(\ell)} = \Theta_L(1)$.
- **Desideratum 5.** Each residual block contributes $\Theta(1/L)$ to feature evolution, so that the total feature change over L blocks is $\Theta_L(1)$.

As for the width (§2.3), $x = \Theta_L(1)$ means that x is constant with respect to the depth L. As shown in previous work (Yang et al., 2023; Bordelon et al., 2023), the hidden activations (Eq. 19) explode—and so Desideratum 4 is violated—unless $\alpha \geq 1/2$ (see §A.5.1). Further, one can show that $\alpha = 1/2$ uniquely satisfies Desideratum 5. As for the width (Theorem 1), this also turns out to be the critical scaling that satisfies all the depth desiderata for PC.

Theorem 2 (Width- and depth-stable feature-learning parameterisations for linear PCNs.)**.** *Consider any width-stable and feature-learning parameterisation of linear residual networks (Eq. 15) with an additional depth scaling exponent α (Eqs. 7-8, 19 & 10). Assume PCNs that learn on the equilibrated energy, and consider joint width-depth limits $N, L \to \infty$ where $L/N \to 0$. Then, the parameterisation satisfying the depth Desiderata for PC is the same as for BP, namely $\alpha = 1/2$.*

The result is proved in §A.5. Similar to the width case, the forward pass desideratum (4) is the same for BP and PC.

4.1. PC convergence to BP for linear residual networks when $L/N \to 0$

We can further show that under any width- *and* depth-stable feature-learning parameterisation of linear residual networks, PC converges to BP in any limit where the width is much larger than the depth, $L/N \to 0$.

Corollary 4.2 (PC convergence to BP on increasingly wider than deep linear residual networks.)**.** *Under any width- and depth-stable feature-learning parameterisation of linear residual networks (Eq. 15) with $\alpha = 1/2$, in any joint width-depth limit $N, L \to \infty$ satisfying $L/N \to 0$, the gradient of the equilibrated energy converges to the BP gradient of the MSE loss. More precisely,*

$$\frac{\|\nabla_{\boldsymbol{\theta}} \mathcal{F}^* - \nabla_{\boldsymbol{\theta}} \mathcal{L}\|_2}{\|\nabla_{\boldsymbol{\theta}} \mathcal{L}\|_2} = O\left(\frac{L}{N}\right), \tag{20}$$

provided the MSE gradient remains non-vanishing.

The result is derived in §A.6. For intuition, note that, in the infinite width and depth limit, the equilibrated energy rescaling of linear residual networks (Eq. 106) scales as

$$s(\boldsymbol{\theta}) = 1 + \Theta\left(L/N\right), \quad \text{as } N, L \to \infty. \tag{21}$$

This scaling is empirically verified in Figures 3 and A.9. Similar to the MLP case, as long as $L/N \to 0$, (i) the rescaling approaches one, (ii) the equilibrated energy converges to the MSE loss, and (iii) the PC gradients converge to the BP gradients. All these results are verified for both linear and nonlinear residual networks (e.g. see Figures 1 & A.11). We note that Corollary 4.2 provides a non-trivial generalisation of a result of Innocenti et al. (2025), which proved only convergence of the objectives at initialisation.

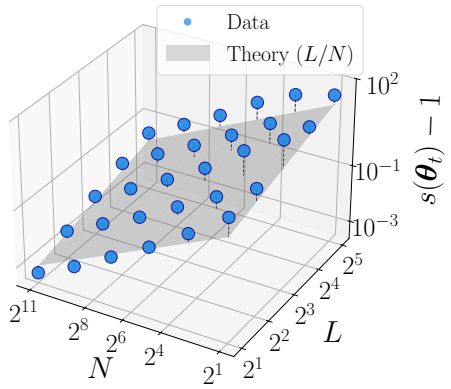


Figure 3. **Empirical verification of Eq. 21.** For linear residual networks trained on CIFAR-10, we plot the empirical equilibrated energy rescaling $s(\boldsymbol{\theta})$ (Eq. 106) minus one as a function of the width N and depth L, against the L/N theoretical prediction (Eq. 21). Note that the same scaling applies to MLPs with infinite depth (Figure A.10), but the stability of their forward pass with depth is more fragile, as discussed in §3.2.

***Takeaway 3**: The set of width- and depth-stable feature-learning parameterisations for PC is the same as for BP. Moreover, under any of these parameterisations, PC converges to BP as long as the width is much larger than the depth.*

5. Experiments with Practical PCNs

Our theoretical analysis makes two important related assumptions: (i) linear networks, and (ii) exact convergence of the PC inference dynamics (Eq. 3). Together, these assumptions allow us to study the equilibrated energy (Eq. 5) as the effective energy on which (linear) PCNs learn. To avoid convergence failure as a potential confounder in testing our theory, all previous experiments performed GD directly on the equilibrated energy. However, in practice we are interested in nonlinear networks, for which there is in

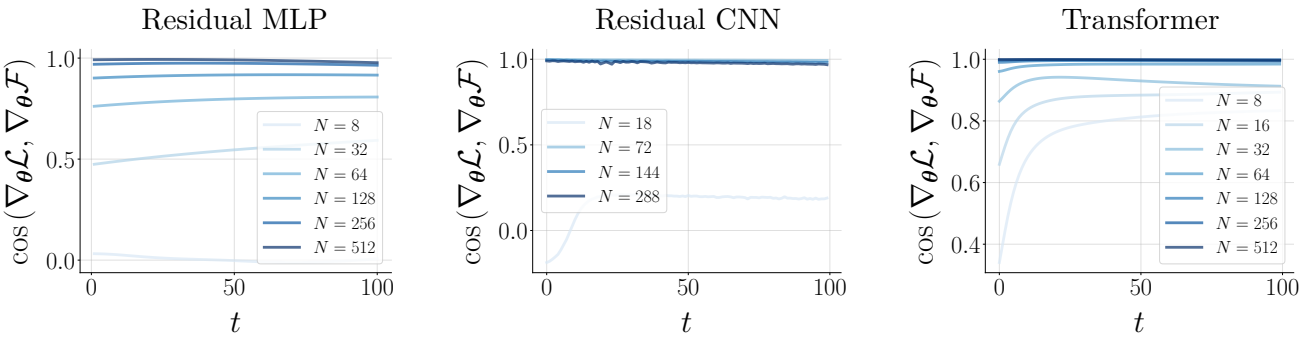


Figure 4. **PC shows high gradient alignment with BP on different *nonlinear architectures* that are much wider than deep, under stable and feature-learning parameterisations.** We trained residual MLPs, CNNs, and (nano-GPT-style) transformers with appropriate extensions of the μP parameterisation (§A.7). The architecture and training details for the residual MLPs were the same as the 32-layer models in Figure 1, with the addition of Tanh as nonlinearity. The residual CNNs had 10 layers, an effective width of $N = k^2 C_{\text{out}}$ with kernel size $k = 3$ and output channels $C_{\text{out}} \in \{2, 8, 16, 32\}$, and were trained on ImageNet-1k using the cross-entropy loss. The transformers had 12 blocks with 8 attention heads, and were trained on character-level Tiny Shakespeare. For more details, see §A.7 and §A.9. Plotted are the cosine similarities for a representative run between the BP loss gradients and the energy gradients at the last step of activity (GD) optimisation (Eq. 3), for different training steps t. Results were similar across seeds.

general no analytical solution of the activities, and inference is performed by some iterative algorithm such as GD (§2.1).

This raises the question: *under stable and feature-learning parameterisations, does PC still converge to BP at large width in practice, i.e. on nonlinear networks when performing inference via optimisation?*

To answer this question, we trained different nonlinear models, including CNNs and (nano-GPT-style) transformers (Radford et al., 2019), recording the cosine similarity between the BP and PC gradients. All experiments used Adam and the mean-field parameterisation, with heuristic extensions for different architectures (see §A.7 & §A.9 for details). Code to reproduce all the results is available at `https://github.com/thebuckleylab/jpc/tree/main/experiments/limits_paper`.

Across all these settings, we find that PC shows almost perfect gradient alignment with BP for much wider than deep models (Figure 4), as long as activity equilibrium seems to be reached. For deeper networks, we found that high alignment held only after a large number of inference steps, in line with previous work (Innocenti et al., 2025; Goemaere et al., 2025). These results suggest that convergence to BP at large width holds under a variety of conditions not directly addressed by our theory, including different nonlinear architectures, optimisers, and loss functions.

> ***Takeaway 4****: Under stable and feature-learning parameterisations, PC still converges to BP on different nonlinear architectures that are much wider than deep, as long as an activity equilibrium seems to be reached.*

6. Discussion

6.1. Summary

Using linear networks, we theoretically derived stable and feature-learning parameterisations for PC when scaling both the model width and depth. Surprisingly, we found that the same set of scalings as for BP ensure stability and non-trivial feature learning for PC (Theorems 1-2; Figure A.5). We further showed that, under any stable parameterisation, the gradients computed by PC converge to the BP gradients when the model is much wider than deep (Corollaries 3.2 & 4.2). Finally, we showed that even in practice, when performing inference via optimisation on nonlinear networks, PC shows high gradient alignment with BP at large width under stable parameterisations, as long as an activity equilibrium seems to be reached (Figure 4).

6.2. Implications

Overall, our results provide hard constraints on the types of parameterisation that are scalable (in model width and depth) with PC. More explicitly, we showed that

> ***if*** *one would like to satisfy the μP desiderata (§2.3 & §4) under our general parameterisation (§2.2), then* ***necessarily*** *μP is the* ***only*** *scalable parameterisation for PC (Theorems 1-2), in the sense of being numerically stable and learning non-trivial features at large width and depth.*

These results go significantly beyond the comparable Corollary 4.3 of Ishikawa et al. (2024), which only showed that there *exists* a parameterisation where linear PCNs implement BP. Also note that Theorems 1-2 imply that PCNs commonly trained in practice are inherently unstable with respect to both the width and depth, since they rely on the SP.

As discussed in §3.2, this includes the unique regime where PC has demonstrated clear benefits over BP in the form of fast saddle-to-saddle dynamics (Innocenti et al., 2024a). However, we cannot necessarily conclude from these results that there exists no other stable and rich parameterisation under which PC does not converge to BP, and we return to this point below.

While the above results may appear negative, they are also the first (to the best of our knowledge) to show—theoretically for linear networks and empirically for nonlinear ones (Figure 4)—that BP can be effectively implemented with a local algorithm (PC) at scale, for models that are much wider than deep.

This result is encouraging for two reasons. First, it aligns with modern large language models, whose width typically exceeds their depth by at least one order of magnitude. This means that, if the convergence of PC inference (Eq. 3) could be substantially accelerated (a point which we address below), then the sequential cost of weight updates could in principle scale as $O(1)$ rather than $O(L)$, since PC weight updates are parallelisable across layers. Second, the brain is arguably much wider than deep (Suzuki et al., 2023), with around $5 - 10$k synapses per neuron and 6 cortical layers. Our results therefore suggest a way in which biology could implement BP with a purely local learning rule.

6.3. Limitations and future directions

Linear theory. While we provided supporting experiments on nonlinear models (Figure 4), linearity remains the main limitation of our theoretical analysis. Building further on Bordelon & Pehlevan (2022b), it could be interesting to investigate whether our theoretical results can be extended to the nonlinear case by using tools from dynamical mean-field theory (DMFT). DMFT allows one to derive a lower-dimensional description of network dynamics in certain limits, and has already been used to gain insights into infinite-width networks trained with other biologically plausible rules such as feedback alignment and error-modulated Hebbian learning (Bordelon & Pehlevan, 2022a). These algorithms do not have alternating activity and weight dynamics like PC, but one could imagine deriving DMFT equations for both optimisation timescales.[3]

Alternative parameterisations. Another interesting direction would be to investigate other types of parameterisation (§2.2), with at least two goals. First, as mentioned above, there could be other stable and feature-learning parameterisations where PC might not converge to BP (see Figure A.5 for a schematic illustration). Exploring such parameterisations would require generalising our considered parameterisation and/or introducing different desiderata (§2.3 & §4). It could also be useful to consider constraints for stable inference, as well as learning, dynamics. This was in part attempted in Ishikawa et al. (2024) by considering scalings of the layer PC energies.

PC inference cost. The main limitation of PC remains the computational cost of inference convergence (Eq. 3), shared with other bio-plausible algorithms with alternating dynamics such as equilibrium propagation (Scellier & Bengio, 2017). On GPUs, it has been estimated that PC becomes about as fast as BP when the number of inference steps is approximately equal to the model depth (Salvatori et al., 2024). Previous studies have shown that, at least for simple datasets, the number of inference steps can remain close to the depth without sacrificing performance (Pinchetti et al., 2024; Innocenti et al., 2025), despite no longer reaching an equilibrium. PC inference can also be accelerated using BP (Goemaere et al., 2025) and other activity initialisations (Pinchetti et al., 2026).

Setting aside the increased memory cost of most of these methods, we speculate that significant speed-ups for PC inference—and so potential compute savings over BP—are more likely to come from analog hardware implementations that physically realise the network dynamics (Momeni et al., 2025; Aifer et al., 2025; Montanari et al., 2026; Wright et al., 2026). Whether PC can be successfully implemented on analog hardware remains an important open question.

Biologically plausible attention. A limitation of our transformer experiment (Figure 4)—with potential implications for analog hardware (Bacvanski et al., 2025)—is that the self-attention mechanism (Vaswani et al., 2017) is inherently non-local, in that it requires computing pairwise interactions between every token across the input sequence. It could be interesting to investigate attention mechanisms where the softmax is itself the negative gradient of an energy function (Singh & Buckley, 2023). Such mechanisms are closely related to modern Hopfield networks (Krotov & Hopfield, 2016; Ramsauer et al., 2020; Hoover et al., 2023) and have some biologically plausible implementations (Kozachkov et al., 2025; Kafraj et al., 2026).

Training to convergence. All our experiments trained models for only 100 steps. This was partly because our aim was to test the theory and partly due to the high computational cost of PC inference, which for the deepest models required up to 100k steps to converge (see §A.9). It could be interesting, as a proof of concept, to show that PC can train a large-scale model to convergence with BP-competitive performance. This experiment would also allow one to leverage the hyperparameter transfer property of μP (Yang et al., 2021), which we did not validate here but predict would hold in the regime where PC converges to BP.

[3]We thank one of the reviewers for this suggestion.

Acknowledgements

FI and RB acknowledge funding from the Wellcome Trust grant 313955/Z/24/Z. RB acknowledges funding from the Medical Research Council grants UKRI/MR/B000936/1 and MC_UU_00003/1. EMA acknowledges funding by UM6P.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Aifer, M., Belateche, Z., Bramhavar, S., Camsari, K. Y., Coles, P. J., Crooks, G., Durian, D. J., Liu, A. J., Marchenkova, A., Martinez, A. J., et al. Solving the compute crisis with physics-based asics. *arXiv preprint arXiv:2507.10463*, 2025. 9

Alonso, N., Millidge, B., Krichmar, J., and Neftci, E. O. A theoretical framework for inference learning. *Advances in Neural Information Processing Systems*, 35:37335–37348, 2022. 1

Alonso, N., Krichmar, J., and Neftci, E. Understanding and improving optimization in predictive coding networks. *arXiv preprint arXiv:2305.13562*, 2023. 1

Amit, D. J., Gutfreund, H., and Sompolinsky, H. Statistical mechanics of neural networks near saturation. *Annals of physics*, 173(1):30–67, 1987. 14

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 32

Bacvanski, M. G., You, X., Hopfield, J., and Krotov, D. Dense associative memories with analog circuits. *arXiv preprint arXiv:2512.15002*, 2025. 9

Bogacz, R. A tutorial on the free-energy framework for modelling perception and learning. *Journal of mathematical psychology*, 76:198–211, 2017. 3

Bordelon, B. and Pehlevan, C. The influence of learning rule on representation dynamics in wide neural networks. *arXiv preprint arXiv:2210.02157*, 2022a. 2, 9

Bordelon, B. and Pehlevan, C. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35: 32240–32256, 2022b. 1, 3, 4, 5, 9, 15, 16, 18, 32, 33

Bordelon, B. and Pehlevan, C. Deep linear network training dynamics from random initialization: Data, width, depth, and hyperparameter transfer. *arXiv preprint arXiv:2502.02531*, 2025. 3

Bordelon, B., Noci, L., Li, M. B., Hanin, B., and Pehlevan, C. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. *arXiv preprint arXiv:2309.16620*, 2023. 1, 6, 7, 15, 23, 29, 32

Bordelon, B., Chaudhry, H., and Pehlevan, C. Infinite limits of multi-head transformer dynamics. *Advances in Neural Information Processing Systems*, 37:35824–35878, 2024. 28

Buckley, C. L., Kim, C. S., McGregor, S., and Seth, A. K. The free energy principle for action and perception: A mathematical review. *Journal of mathematical psychology*, 81:55–79, 2017. 3

Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019. 15

Dey, N., Zhang, B. C., Noci, L., Li, M., Bordelon, B., Bergsma, S., Pehlevan, C., Hanin, B., and Hestness, J. Don't be lazy: Completep enables compute-efficient deep transformers. *arXiv preprint arXiv:2505.01618*, 2025. 1, 15, 28, 29

Friston, K. A theory of cortical responses. *Philosophical transactions of the Royal Society B: Biological sciences*, 360(1456):815–836, 2005. 1

Goemaere, C., Oliviers, G., Bogacz, R., and Demeester, T. epc: Fast and deep predictive coding in digital simulation. *arXiv preprint arXiv:2505.20137*, 2025. 1, 8, 9, 15

Hardt, M. and Ma, T. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016. 6

Hayou, S. and Yang, G. Width and depth limits commute in residual networks. In *International Conference on Machine Learning*, pp. 12700–12723. PMLR, 2023. 15, 23

Hayou, S., Clerico, E., He, B., Deligiannidis, G., Doucet, A., and Rousseau, J. Stable resnet. In *International Conference on Artificial Intelligence and Statistics*, pp. 1324–1332. PMLR, 2021. 23

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015. 4, 17

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal

large language models. *arXiv preprint arXiv:2203.15556*, 2022. 14

Hoover, B., Liang, Y., Pham, B., Panda, R., Strobelt, H., Chau, D. H., Zaki, M., and Krotov, D. Energy transformer. *Advances in neural information processing systems*, 36: 27532–27559, 2023. 9

Innocenti, F., Singh, R., and Buckley, C. Understanding predictive coding as a second-order trust-region method. In *ICML Workshop on Localized Learning (LLW)*, 2023. 1

Innocenti, F., Achour, E. M., Singh, R., and Buckley, C. L. Only strict saddles in the energy landscape of predictive coding networks? *Advances in Neural Information Processing Systems*, 37:53649–53683, 2024a. 1, 3, 6, 9, 30, 32

Innocenti, F., Kinghorn, P., Yun-Farmbrough, W., Varona, M. D. L., Singh, R., and Buckley, C. L. Jpc: Flexible inference for predictive coding networks in jax. *arXiv preprint arXiv:2412.03676*, 2024b. 32

Innocenti, F., Achour, E. M., and Buckley, C. L. μpc: Scaling predictive coding to 100+ layer networks. *arXiv preprint arXiv:2505.13124*, 2025. 1, 3, 7, 8, 9, 15, 16, 24

Ishikawa, S., Yokota, R., and Karakida, R. Local loss optimization in the infinite width: Stable parameterization of predictive coding networks and target propagation. *arXiv preprint arXiv:2411.02001*, 2024. 2, 3, 6, 8, 9, 15, 18

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018. 5, 15, 16

Jacot, A., Ged, F., Şimşek, B., Hongler, C., and Gabriel, F. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. *arXiv preprint arXiv:2106.15933*, 2021. 6

Kafraj, M. S., Krotov, D., and Latham, P. E. A biologically plausible dense associative memory with exponential capacity. *arXiv preprint arXiv:2601.00984*, 2026. 9

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 14

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4, 28

Kozachkov, L., Slotine, J.-J., and Krotov, D. Neuron–astrocyte associative memory. *Proceedings of the National Academy of Sciences*, 122(21):e2417788122, 2025. 9

Krotov, D. and Hopfield, J. J. Dense associative memory for pattern recognition. *Advances in neural information processing systems*, 29, 2016. 9

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–50. Springer, 2002. 4, 17

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015. 1

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017. 15

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019. 15

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020. 1

Matthews, A. G. d. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018. 15

Millidge, B., Song, Y., Salvatori, T., Lukasiewicz, T., and Bogacz, R. Backpropagation at the infinitesimal inference limit of energy-based models: Unifying predictive coding, equilibrium propagation, and contrastive hebbian learning. *arXiv preprint arXiv:2206.02629*, 2022a. 1, 15

Millidge, B., Tschantz, A., and Buckley, C. L. Predictive coding approximates backprop along arbitrary computation graphs. *Neural Computation*, 34(6):1329–1368, 2022b. 1, 15

Momeni, A., Rahmani, B., Scellier, B., Wright, L. G., McMahon, P. L., Wanjura, C. C., Li, Y., Skalli, A., Berloff, N. G., Onodera, T., et al. Training of physical neural networks. *Nature*, 645(8079):53–61, 2025. 9

Montanari, A. N., Bullo, F., Krotov, D., and Motter, A. E. Energy-based dynamical models for neurocomputation, learning, and optimization. *arXiv preprint arXiv:2604.05042*, 2026. 9

Neal, R. M. Priors for infinite networks. In *Bayesian learning for neural networks*, pp. 29–53. Springer, 1996. 15

Pinchetti, L., Qi, C., Lokshyn, O., Olivers, G., Emde, C., Tang, M., M'Charrak, A., Frieder, S., Menzat, B., Bogacz, R., et al. Benchmarking predictive coding networks–made simple. *arXiv preprint arXiv:2407.01163*, 2024. 1, 9

Pinchetti, L., Frieder, S., Lukasiewicz, T., and Salvatori, T. Faster predictive coding networks via better initialization. *arXiv preprint arXiv:2601.20895*, 2026. 9

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016. 6, 18, 24

Qi, C., Forasassi, M., Lukasiewicz, T., and Salvatori, T. Towards the training of deeper predictive coding neural networks. *arXiv preprint arXiv:2506.23800*, 2025. 1

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 8

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020. 9

Rao, R. P. and Ballard, D. H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999. 1

Rosenbaum, R. On the relationship between predictive coding and backpropagation. *Plos one*, 17(3):e0266102, 2022. 1, 15

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 1

Salvatori, T., Song, Y., Lukasiewicz, T., Bogacz, R., and Xu, Z. Predictive coding can do exact backpropagation on convolutional and recurrent neural networks. *arXiv preprint arXiv:2103.03725*, 2021. 1, 15

Salvatori, T., Song, Y., Yordanov, Y., Millidge, B., Sha, L., Emde, C., Xu, Z., Bogacz, R., and Lukasiewicz, T. A stable, fast, and fully automatic learning algorithm for predictive coding networks. In *International Conference on Learning Representations*, volume 2024, pp. 19607–19631, 2024. 9

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. 6

Scellier, B. and Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Frontiers in computational neuroscience*, 11:24, 2017. 9

Schoenholz, S. S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016. 6, 18, 24

Singh, R. and Buckley, C. L. Attention as implicit structural inference. *Advances in Neural Information Processing Systems*, 36:24929–24946, 2023. 9

Song, Y., Lukasiewicz, T., Xu, Z., and Bogacz, R. Can the brain do backpropagation?—exact implementation of backpropagation in predictive coding networks. *Advances in neural information processing systems*, 33: 22566–22579, 2020. 1, 15

Song, Y., Millidge, B., Salvatori, T., Lukasiewicz, T., Xu, Z., and Bogacz, R. Inferring neural activity before plasticity as a foundation for learning beyond backpropagation. *Nature neuroscience*, 27(2):348–358, 2024. 1

Suzuki, M., Pennartz, C. M., and Aru, J. How deep is the brain? the shallow brain hypothesis. *Nature Reviews Neuroscience*, 24(12):778–791, 2023. 9

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 9, 29

Whittington, J. C. and Bogacz, R. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5):1229–1262, 2017. 1, 15

Wright, L. G., Wang, T., Onodera, T., and McMahon, P. L. Physical foundation models: Fixed hardware implementations of large-scale neural networks. *arXiv preprint arXiv:2604.27911*, 2026. 9

Yang, G. and Hu, E. J. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020. 4, 5, 15

Yang, G. and Hu, E. J. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pp. 11727–11737. PMLR, 2021. 1, 4, 5, 15, 16, 33

Yang, G. and Littwin, E. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv preprint arXiv:2308.01814*, 2023. 28

Yang, G., Hu, E., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tuning

large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021. 1, 2, 9, 15

Yang, G., Yu, D., Zhu, C., and Hayou, S. Tensor programs vi: Feature learning in infinite-depth neural networks. *arXiv preprint arXiv:2310.02244*, 2023. 6, 7, 15, 23

A. Appendix

Table of Contents

A.1 General notation . . . 14

A.2 Related work . . . 14

A.3 Table of width-aware parameterisations for BP . . . 16

A.4 Derivation of width scalings for linear PCNs . . . 16

- **A.4.1** Desideratum 1: Preactivations are $\Theta_N(1)$ at initialisation . . . 17
- **A.4.2** Useful asymptotic results for MLPs . . . 17
- **A.4.3** Desideratum 2: Network predictions evolve in $\Theta_N(1)$ time during training . . . 19
- **A.4.4** Desideratum 3: Features evolve in $\Theta_N(1)$ time during training . . . 21
- **A.4.5** Combining constraints . . . 23

A.5 Derivation of depth scalings for linear residual PCNs . . . 23

- **A.5.1** Desideratum 4: Preactivations are $\Theta_L(1)$ at initialisation . . . 23
- **A.5.2** Useful asymptotic results for residual networks . . . 24
- **A.5.3** Desideratum 5: Each residual block has a $\Theta(1/L)$ contribution to feature evolution . . . 25

A.6 Convergence of the equilibrated energy gradients to the MSE gradients . . . 26

A.7 Heuristic extension to other architectures and optimisers . . . 28

- **A.7.1** CNNs . . . 28
- **A.7.2** Transformers . . . 29

A.8 Additional experiments . . . 30

- **A.8.1** Nonlinear networks . . . 30
- **A.8.2** Learning regimes . . . 30

A.9 Experimental details . . . 32

A.10 Compute resources . . . 32

A.11 Supplementary figures . . . 33

A.1. General notation

Vectors $\boldsymbol{v}$ and matrices $\boldsymbol{A}$ are typed in bold, while scalars c and C are in non-bold, including element-wise indexing of vectors v_i and matrices A_{ij}. All vectors are column-oriented unless otherwise stated (e.g. the output weights $\boldsymbol{w}^{(L)} \in \mathbb{R}^{1\times N}$), and the dot product between two vectors is denoted as $\boldsymbol{u} \cdot \boldsymbol{v}$. $\|\cdot\|$ is the l_2 or Euclidean norm, $\|\cdot\|_F$ is the Frobenius norm, and $\mathrm{vec}(\cdot)$ is the row-vector operator. We will denote expectations with respect to model parameters $\boldsymbol{\theta}$ with angle brackets: $\langle\cdot\rangle := \mathbb{E}_{\boldsymbol{\theta}}[\cdot]$. $\boldsymbol{I}_N$ is the identity matrix of size N. The notation $x = \Theta_N(1)$ indicates that the absolute value of x is bounded away from both zero and infinity as $N \to \infty$, i.e. $0 < \liminf_{N\to\infty} |x(N)| \leq \limsup_{N\to\infty} |x(N)| < \infty$.

A.2. Related work

Scaling limits of neural networks. There is a long history of studying neural networks by taking infinite limits of several coarse (thermodynamic) variables (Amit et al., 1987), such as the number of data samples or the size of the model. Limits taken with respect to model size are the most relevant for our purposes and have also had great practical impact.[4]

[4] "Scaling laws", where model size, training time and dataset size are all scaled, have arguably been the most impactful (Kaplan et al., 2020; Hoffmann et al., 2022). However, in contrast to the theoretical limits reviewed above, scaling laws originated as empirical observations.

Early results on the infinite-width limit. The correspondence between Gaussian Processes and the function computed by feedforward networks at initialisation (NNGP) was one of the first results on the infinite-width limit (Neal, 1996; Lee et al., 2017; Matthews et al., 2018). Jacot et al. (2018) then famously showed that the gradient descent training dynamics of wide networks are equivalent to a kernel method termed the "neural tangent kernel" (NTK). However, in this regime, the parameters stay close to their initialisation, and the network features or preactivations barely move at large width, resulting in "lazy" learning (Chizat et al., 2019; Lee et al., 2019).

The mean-field/μP parameterisation. Motivated by the lack of feature learning in the NTK regime, Yang & Hu (2020) derived a parameterisation in the infinite-width limit where the feature updates are "maximal" in a well-defined sense (see Desideratum 3 in §2.3)—hence the "maximal update parameterisation" or μP. Moreover, Yang et al. (2021) showed that under μP, hyperparameters such as the learning rate also remain stable across model widths, enabling zero-shot hyperparameter transfer. While Yang & Hu (2020) derived the theoretical limit using the "Tensor Programs" framework, Bordelon & Pehlevan (2022b) later described the same limit using a dynamical mean field theory inspired from statistical physics. These and other parameterisations are explained in more detail in §3 and precisely defined in Table 2.

Depth-aware parameterisations. μP has been extended to a variety of architectures and optimisers. The extensions to infinitely deep residual networks (Yang & Hu, 2020; Bordelon et al., 2023; Dey et al., 2025), for which the order of the width and depth limits does not matter (i.e. it commutes) (Hayou & Yang, 2023), are most relevant for our purposes. As reviewed in §4, such parameterisations mainly introduce a depth-dependent rescaling $L^{-\alpha}$ of the residual branches and enable hyperparameter transfer across model depths.

Reparameterisations of PCNs. Our work is most closely related to Ishikawa et al. (2024) and Innocenti et al. (2025). Ishikawa et al. (2024) derived an infinite-width, μP-like parameterisation for linear PCNs. The key difference with our work is that Ishikawa et al. (2024) showed the *existence* of a parameterisation where linear PCNs implement BP with GD (see their Corollary 4.3). By contrast, we prove a much stronger result: if one would like to satisfy the μP desiderata, then *necessarily*, μP is the only stable and feature-learning parameterisation for PC (Theorems 1-2). Our work also differs from Ishikawa et al. (2024) in the following respects: (i) we derive a slightly more general set of stable and feature-learning parameterisations, in the same way as Bordelon & Pehlevan (2022b) recovered μP (Yang & Hu, 2021); (ii) we extend the parameterisation to depth; and (iii) we do not include scalings of the layer PC energies or activity learning rates (Eq. 2).

Innocenti et al. (2025), on the other hand, used a parameterisation inspired by depth extensions of μP (with $\alpha = 1/2$) (Yang et al., 2023; Bordelon et al., 2023) to stabilise the forward pass of deep PCNs ("μPC"). Notably, they found that this parameterisation enabled the training of 100+ layer residual networks on standard classification benchmarks with hyperparameter transfer, at competitive performance with BP. However, as noted in §4, beyond the forward pass stability, it remains unclear whether this parameterisation satisfies the depth feature-learning desideratum (5) for PC. We clearly answer this question in the affirmative in §4 with Theorem 2.

It is also interesting to note that Innocenti et al. (2025) used Adam for their experiments but did not rescale the learning rate according to the theoretical prescriptions (see §A.7), noting that such scalings led to unstable training. This could be due to the fact that they failed to reach convergence of the PC inference dynamics under this parameterisation (see §5). Innocenti et al. (2025) also proved a convergence result to BP similar to our Corollary 4.2, which we discuss below.

Finally, Goemaere et al. (2025) proposed a reparameterisation of PCNs where errors, instead of activities, are optimised with BP during inference ("ePC"). Since this reparameterisation changes the network's forward pass and the standard PC energy (Eq. 2), it remains unclear whether ePC is stable and rich with respect to both width and depth—and so whether it is fully scalable. This could be determined by extending our analysis to the ePC energy.

Previous correspondences between PC and BP. Early research on PC as a learning algorithm established two main conditions under which PC can approximate or compute exactly the same gradients as BP. First, as also briefly reviewed in §3, PC approximates BP when the output energy is much larger than the other layer energies (Whittington & Bogacz, 2017), which occurs when the equilibrium of the activities is close to the forward pass. Second, if the weight updates are coordinated with specific inference steps, then PC exactly implements BP (Song et al., 2020). Various extensions and generalisations of these results have been formulated (Millidge et al., 2022b; Rosenbaum, 2022; Salvatori et al., 2021; Millidge et al., 2022a).

More recently, Innocenti et al. (2025) proved convergence, *at initialisation*, of the equilibrated energy to the MSE loss under

the same depth parameterisation of linear residual networks considered in §4. Our Corollary 4.2 shows that this result holds throughout training. We also note that, unlike our results, Innocenti et al. (2025) found that the empirical correspondence between the equilibrated energy and the MSE loss deteriorated with training at large width and depth. This could have been due to numerical instabilities in their computation of the equilibrated energy, which was based on inverting a highly ill-conditioned Hessian matrix.

A.3. Table of width-aware parameterisations for BP

The Table below shows the scalings defining the width-aware parameterisations reviewed in §2.3.

Table 2. Summary of (a_ℓ, b_ℓ, c, d) parameterisations derived for BP at large width. For a description of the width-scaling exponents, see Table 1. The standard parameterisation (SP) includes all common frameworks (e.g. PyTorch) that initialise weights with variance inversely proportional to their input dimension. The NTK parameterisation is from Jacot et al. (2018), while the mean-field and μP parameterisations refer to those introduced in Bordelon & Pehlevan (2022b) and Yang & Hu (2021), respectively. Note that we omit any data-dependent normalisation (e.g. $1/\sqrt{D}$) of the first layer's preactivations (see Eq. 10).

Exponent	**SP**	**NTK**	**Mean-field**	**μP**
a_1	0	0	0	$-1/2$
b_1	0	0	0	1
$a_\ell, \quad \ell \in \{2, \dots, L\}$	0	$1/2$	$1/2$	0
$b_\ell, \quad \ell \in \{2, \dots, L\}$	1	0	0	1
c	0	0	0	1
d	0	0	$1/2$	$1/2$

A.4. Derivation of width scalings for linear PCNs

Based on the general (a_ℓ, b_ℓ, c, d) parameterisation of linear MLPs defined in §2.2 and the equilibrated energy of linear PCNs (Eq. 5), this section derives the width scaling exponents for PC that satisfy the 3 Desiderata listed in §2.3, proving Theorem 1. For convenience, we recall our MLP parameterisation:

$$f_\mu = \frac{1}{\gamma} h_\mu^{(L)}, \tag{22}$$

$$h_\mu^{(L)} = \frac{1}{N^{a_L}} \boldsymbol{w}^{(L)} \boldsymbol{h}_\mu^{(L-1)}, \qquad w_i^{(L)} \sim \mathcal{N}\left(0, \frac{1}{N^{b_L}}\right), \tag{23}$$

$$\boldsymbol{h}_\mu^{(\ell)} = \frac{1}{N^{a_\ell}} \boldsymbol{W}^{(\ell)} \boldsymbol{h}_\mu^{(\ell-1)}, \qquad W_{ij}^{(\ell)} \sim \mathcal{N}\left(0, \frac{1}{N^{b_\ell}}\right), \tag{24}$$

$$\boldsymbol{h}_\mu^{(1)} = \frac{1}{N^{a_1}\sqrt{D}} \boldsymbol{W}^{(1)} \boldsymbol{x}_\mu, \qquad W_{ij}^{(1)} \sim \mathcal{N}\left(0, \frac{1}{N^{b_1}}\right), \tag{25}$$

where again recall that $\gamma = \gamma_0 N^d$. Under this parameterisation, the equilibrated PC energy (Eq. 5) is given by

$$\mathcal{F}^*(\boldsymbol{\theta}) = \frac{1}{s(\boldsymbol{\theta})} \mathcal{L}(\boldsymbol{\theta}), \tag{26}$$

$$s(\boldsymbol{\theta}) = 1 + \gamma^{-2} \sum_{\ell=2}^{L} N^{-2\sum_{k=\ell}^{L} a_k} \|\boldsymbol{W}^{(L:\ell)}\|^2. \tag{27}$$

For what follows, it will also be useful to define the gradient flow on the equilibrated energy (Eq. 5)

$$\frac{d\boldsymbol{\theta}}{dt} = -\eta \frac{\partial \mathcal{F}^*}{\partial \boldsymbol{\theta}} = -\eta \left(\frac{1}{s(\boldsymbol{\theta})} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} - \frac{\mathcal{L}}{s(\boldsymbol{\theta})^2} \frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right), \tag{28}$$

which decomposes into two terms: (i) a rescaled loss gradient, and (ii) an extra term depending on the parameter derivative of the rescaling. These terms will appear in the analysis of the second and third desiderata (§A.4.3-A.4.4). Also recall that $\eta = \eta_0 \gamma^2 N^{-c}$. The following subsections derive specific constraints for PC for each desideratum listed in §2.3. Our notation and derivation style closely follow those of Bordelon & Pehlevan (2022b), with a physics-level rigour of analysis.

A.4.1. Desideratum 1: Preactivations are $\Theta_N(1)$ at initialisation

Verification of this desideratum does not differ between BP and PC, since the forward pass is algorithm-independent. We emphasise that the following derivation is not novel, and similar calculations can in fact be traced to classic weight initialisation schemes (LeCun et al., 2002; He et al., 2015). However, we include it for completeness.

Starting from the activations of the first layer, note that $\boldsymbol{h}_\mu^{(1)}$ is a sum of Gaussian random variables (Eq. 10). Its statistics are therefore fully specified by its mean and covariance:

$$\left\langle h_{\mu,i}^{(1)} \right\rangle = \frac{1}{N^{a_1}\sqrt{D}} \sum_{k=1}^{D} \left\langle W_{ik}^{(1)} \right\rangle x_{\mu,k} = 0, \tag{29}$$

$$\left\langle h_{\mu,i}^{(1)} h_{\nu,j}^{(1)} \right\rangle = \frac{1}{N^{2a_1} D} \sum_{k=1}^{D} \sum_{k'=1}^{D} \left\langle W_{ik}^{(1)}(0) W_{jk'}^{(1)}(0) \right\rangle x_{\mu,k} x_{\nu,k'} \tag{30}$$

$$= \frac{1}{N^{2a_1} D} \sum_{k=1}^{D} \sum_{k'=1}^{D} \delta_{ij}\delta_{kk'} \frac{1}{N^{b_1}} x_{\mu,k} x_{\nu,k'} \tag{31}$$

$$= \delta_{ij} \frac{1}{N^{2a_1+b_1} D} \sum_{k=1}^{D} x_{\mu,k} x_{\nu,k}, \tag{32}$$

where $W_{ij}^{(1)}(0)$ denotes the value of a weight at initialisation ($t = 0$). To keep the covariance $\Theta_N(1)$ as $N \to \infty$, we therefore need the constraint $2a_1 + b_1 = 0$. Moving on to the next layer, its mean and covariance are given by

$$\left\langle h_{\mu,i}^{(2)} \right\rangle = \frac{1}{N^{a_\ell}} \sum_{k=1}^{N} \left\langle W_{ik}^{(2)}(0) \right\rangle \left\langle h_{\mu,k}^{(1)} \right\rangle = 0, \tag{33}$$

$$\left\langle h_{\mu,i}^{(2)} h_{\nu,j}^{(2)} \right\rangle = \frac{1}{N^{2a_\ell}} \sum_{k=1}^{N} \sum_{k'=1}^{N} \left\langle W_{ik}^{(2)}(0) W_{jk'}^{(2)}(0) \right\rangle \left\langle h_{\mu,k}^{(1)} h_{\nu,k'}^{(1)} \right\rangle \tag{34}$$

$$= \frac{1}{N^{2a_\ell}} \sum_{k=1}^{N} \sum_{k'=1}^{N} \delta_{ij}\delta_{kk'} \frac{1}{N^{b_\ell}} \left\langle h_{\mu,k}^{(1)} h_{\nu,k'}^{(1)} \right\rangle \tag{35}$$

$$= \delta_{ij} \frac{1}{N^{2a_\ell+b_\ell}} \sum_{k=1}^{N} \left\langle h_{\mu,k}^{(1)} h_{\nu,k}^{(1)} \right\rangle \tag{36}$$

$$= \delta_{ij} \frac{1}{N^{2a_\ell+b_\ell-1}} \left\langle h_{\mu,1}^{(1)} h_{\nu,1}^{(1)} \right\rangle, \tag{37}$$

where in the last step we use the fact that neurons $h_{\mu,k}^{(1)}$ are i.i.d. across k, and so the average over k equals the value for any fixed k, such as $k = 1$. This argument can be generalised inductively to deeper layers, and so overall, we obtain the following constraints for Desideratum 1

$$\boxed{\begin{aligned} 2a_1 + b_1 &= 0, \\ 2a_\ell + b_\ell &= 1. \end{aligned}} \tag{38}$$

Note that the constraint for the first layer differs from that of other layers because it is the only layer whose input dimension does not grow to infinity (as the dataset is assumed to be constant with respect to the width). We will find an analogous result for the next desideratum. Also note that, to satisfy this desideratum, one can choose to scale the initialisation variance of the weights (as in the SP) or the preactivations themselves (as in the NTK and μP parameterisations), or both if one wishes.

A.4.2. Useful asymptotic results for MLPs

It is useful, at this point, to present a number of asymptotic results that will simplify the analysis of the subsequent desiderata. These results concern the following objects:

- "feature and gradient kernels" (defined below);

- the derivative of the network function with respect to the weights of any layer, $\partial f / \partial \boldsymbol{W}^{(\ell)}$;
- the equilibrated energy rescaling $s(\boldsymbol{\theta})$; and
- the derivative of the equilibrated energy rescaling with respect to layer weights, $\partial s / \partial \boldsymbol{W}^{(\ell)}$.

Note that the first two objects have been studied for BP, while the last two are specific to PC. In §A.5.2, we will extend the last two results to linear residual networks.

Feature and gradient kernels. First, following Bordelon & Pehlevan (2022b) we define the "feature kernels"

$$\Phi^{(\ell)}_{\mu\nu} \coloneqq \frac{1}{N}\sum_{k=1}^{N} h^{(\ell)}_{\mu,k} h^{(\ell)}_{\nu,k} = \frac{1}{N}\boldsymbol{h}^{(\ell)}_{\mu}\cdot\boldsymbol{h}^{(\ell)}_{\nu}, \quad \ell \in \{1,\dots,L-1\}, \tag{39}$$

$$\Phi^{(0)}_{\mu\nu} \coloneqq \frac{1}{D}\boldsymbol{x}_{\mu}\cdot\boldsymbol{x}_{\nu}, \tag{40}$$

as well as the "gradient kernels"

$$G^{(\ell)}_{\mu\nu} \coloneqq \frac{1}{N}\sum_{k=1}^{N} g^{(\ell)}_{\mu,k} g^{(\ell)}_{\nu,k} = \frac{1}{N}\boldsymbol{g}^{(\ell)}_{\mu}\cdot\boldsymbol{g}^{(\ell)}_{\nu}, \quad \boldsymbol{g}^{(\ell)}_{\mu} = \sqrt{N}\frac{\partial h^{(L)}_{\mu}}{\partial \boldsymbol{h}^{(\ell)}_{\mu}}, \tag{41}$$

$$G^{(L)}_{\mu\nu} = 1. \tag{42}$$

These objects (and their nonlinear versions) have been previously studied in the signal propagation literature (Poole et al., 2016; Schoenholz et al., 2016) under the names of forward and backward kernels, respectively. As shown by Bordelon & Pehlevan (2022b), under the considered parameterisation, both the forward and backward kernels concentrate around their expectation as $N \to \infty$ and are both width-stable:

$$\left\langle \Phi^{(\ell)}_{\mu\nu} \right\rangle = \frac{1}{N}\sum_{k=1}^{N}\left\langle h^{(\ell)}_{\mu,k} h^{(\ell)}_{\nu,k}\right\rangle = \Theta_N(1), \tag{43}$$

$$\left\langle G^{(\ell)}_{\mu\nu} \right\rangle = \frac{1}{N}\sum_{k=1}^{N}\left\langle g^{(\ell)}_{\mu,k} g^{(\ell)}_{\nu,k}\right\rangle = \Theta_N(1). \tag{44}$$

Derivative of the network function with respect to layer weights. It will also be useful to characterise the order of the parameter derivative of the network function with respect to N

$$\frac{\partial f_{\mu}}{\partial W^{(\ell)}_{ij}} = \frac{1}{\gamma}\frac{\partial h^{(L)}_{\mu}}{\partial W^{(\ell)}_{ij}} = \frac{1}{\gamma\sqrt{N}}\underbrace{\left(\sqrt{N}\frac{\partial h^{(L)}_{\mu}}{\partial h^{(\ell)}_{\mu,i}}\right)}_{g^{(\ell)}_i}\frac{\partial h^{(\ell)}_{\mu,i}}{\partial W^{(\ell)}_{ij}} \tag{45}$$

$$= \gamma_0^{-1} N^{-d-1/2} g^{(\ell)}_i \frac{1}{N^{a_\ell}} h^{(\ell-1)}_{\mu,j} = \Theta(N^{-d-a_\ell-1/2}). \tag{46}$$

Equilibrated energy rescaling (Eq. 27). For many of the subsequent analyses, it will be useful to understand the asymptotic behaviour of the equilibrated energy rescaling under our parameterisation, with both the network width N and depth L. As shown in Ishikawa et al. (2024), by the law of large numbers, the squared norm term in the equilibrated energy rescaling (Eq. 27) also converges in the infinite-width limit, giving

$$\left\langle \|\boldsymbol{W}^{(L:\ell)}\|^2 \right\rangle = \left\langle \|\boldsymbol{w}^{(L)}\boldsymbol{W}^{(L-1)}\cdots\boldsymbol{W}^{(\ell)}\|^2 \right\rangle = N^{1-b_L}\prod_{k=\ell}^{L-1} N^{1-b_k} = N^{\sum_{k=\ell}^{L}(1-b_k)}. \tag{47}$$

For intuition, this is easier to see for the one-hidden-layer case where $\left\langle \|\boldsymbol{w}^{(L)}\|^2 \right\rangle = \left\langle \sum_{k=1}^{N}(w^{(L)}_k)^2 \right\rangle = N^{1-b_L}$. If we plug in the forward pass constraint $b_\ell = 1 - 2a_\ell$ from the first desideratum (§A.4.1), Eq. 47 simplifies to $N^{\sum_{k=\ell}^{L}(1-b_k)} =$

$N^{2\sum_{k=\ell}^{L} a_k}$. This leads to the useful result

$$\sum_{\ell=2}^{L} N^{-2\sum_{k=\ell}^{L} a_k} \|\boldsymbol{W}^{(L:\ell)}\|^2 = \sum_{\ell=2}^{L} N^{-2\sum_{k=\ell}^{L} a_k} N^{2\sum_{k=\ell}^{L} a_k} = \Theta_N(1), \tag{48}$$

which we will rely on in many subsequent derivations. We emphasise that this result assumes the forward pass constraint from the first desideratum (which is what leads to the cancellation of the scaled norm terms). For the later depth-scaling analysis (§A.5), it will be important to realise that, if the depth is not constant with respect to the width, then Eq. 48 becomes $\Theta(L)$. If one assumes growing width *and* depth, we therefore get the following result:

$$s(\boldsymbol{\theta}) = 1 + \Theta(\gamma^{-2} L) = 1 + \Theta(L/N^{2d}), \quad N, L \to \infty. \tag{49}$$

Derivative of the equilibrated energy rescaling. To determine the scaling of $\partial s / \partial \boldsymbol{W}^{(\ell)}$, it is convenient to define the effective weights

$$\overline{\boldsymbol{W}}^{(m)} := N^{-a_m} \boldsymbol{W}^{(m)}. \tag{50}$$

Under the forward-pass constraint $2a_\ell + b_\ell = 1$, the entries of these matrices are $\Theta_N(N^{-1/2})$. At fixed depth, products of the effective weights have $\Theta_N(1)$ operator norm and $\Theta_N(N^{-1/2})$ entries. The rescaling can therefore be written as

$$s(\boldsymbol{\theta}) = 1 + \gamma^{-2} \sum_{k=2}^{L} \left\| \overline{\boldsymbol{W}}^{(L:k)} \right\|^2. \tag{51}$$

Consider a hidden matrix $\boldsymbol{W}^{(\ell)}$, with $\ell \in \{2, \dots, L-1\}$. Differentiating gives

$$\frac{\partial s}{\partial W_{ij}^{(\ell)}} = 2\gamma^{-2} N^{-a_\ell} \sum_{k=2}^{\ell} \overline{W}_i^{(L:\ell+1)} \left[\overline{\boldsymbol{W}}^{(L:\ell)} \overline{\boldsymbol{W}}^{(\ell-1:k)} \overline{\boldsymbol{W}}^{(\ell-1:k)\top} \right]_j. \tag{52}$$

The first factor in every summand is $\Theta_N(N^{-1/2})$. The vector inside the square brackets has $\Theta_N(1)$ norm and hence a typical component of order $\Theta_N(N^{-1/2})$. Consequently, every summand is $\Theta_N(N^{-1})$. Therefore, at fixed depth we get

$$\frac{\partial s}{\partial W_{ij}^{(\ell)}} = \Theta_N\left(N^{-2d-a_\ell-1}\right), \qquad \ell \in \{2, \dots, L-1\}. \tag{53}$$

The output weights must be treated separately. Since the corresponding vector in the derivative has typical entries $\Theta_N(N^{-1/2})$,

$$\frac{\partial s}{\partial w_i^{(L)}} = O\left(L N^{-2d-a_L-1/2}\right). \tag{54}$$

Finally, s does not depend on the input weights, so $\partial s / \partial \boldsymbol{W}^{(1)} = \boldsymbol{0}$.

A.4.3. Desideratum 2: Network predictions evolve in $\Theta_N(1)$ time during training

This desideratum requires that $df/dt = \Theta_N(1)$. Let us unpack this:

$$\frac{df(\boldsymbol{x}_\mu; \boldsymbol{\theta})}{dt} = \frac{\partial f_\mu}{\partial \boldsymbol{\theta}} \cdot \frac{d\boldsymbol{\theta}}{dt} \tag{55}$$

$$= -\eta \left[\frac{1}{s(\boldsymbol{\theta}) P} \sum_{\nu=1}^{P} \frac{\partial f_\mu}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \mathcal{L}_\nu}{\partial \boldsymbol{\theta}} - \frac{\mathcal{L}_\nu}{s(\boldsymbol{\theta})^2} \frac{\partial f_\mu}{\partial \boldsymbol{\theta}} \cdot \frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right] \tag{56}$$

$$= \underbrace{\eta \frac{1}{s(\boldsymbol{\theta}) P} \sum_{\nu=1}^{P} \frac{\partial f_\mu}{\partial \boldsymbol{\theta}} \cdot \frac{\partial f_\nu}{\partial \boldsymbol{\theta}} \Delta_\nu}_{q} + \underbrace{\eta \frac{\mathcal{L}_\nu}{s(\boldsymbol{\theta})^2} \frac{\partial f_\mu}{\partial \boldsymbol{\theta}} \cdot \frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}}_{r}, \tag{57}$$

where $\Delta_\nu = -\partial \mathcal{L}_\nu / \partial f_\nu$. Note that, in contrast to BP, the evolution of the network function for PC is given by two terms: (i) a rescaled NTK term q, where $\partial f_\mu / \partial \boldsymbol{\theta} \cdot \partial f_\nu / \partial \boldsymbol{\theta}$ is the so-called NTK; and (ii) an extra term r coming from the gradient of the equilibrated energy (see Eq. 28). Below, we analyse the order of these two terms separately. Note that we will make use of the asymptotic results derived in §A.4.2.

Rescaled NTK term q. Let us first recall from our parameterisation that $\eta = \eta_0 \gamma^2 N^{-c}$. Given that the dataset size and loss delta Δ_ν are constants with respect to the network width, the rescaled NTK term q (in Eq. 57) requires the following to be $\Theta_N(1)$:

$$\frac{\gamma^2}{N^c} \frac{1}{s(\boldsymbol{\theta})} \frac{\partial f_\mu}{\partial \boldsymbol{\theta}} \cdot \frac{\partial f_\nu}{\partial \boldsymbol{\theta}} = \frac{1}{N^c s(\boldsymbol{\theta})} \frac{\partial h_\mu^{(L)}}{\partial \boldsymbol{\theta}} \cdot \frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{\theta}} \tag{58}$$

$$= \frac{1}{N^c s(\boldsymbol{\theta})} \sum_{\ell=1}^{L} \sum_{i,j} \frac{\partial h_\mu^{(L)}}{\partial W_{ij}^{(\ell)}} \frac{\partial h_\nu^{(L)}}{\partial W_{ij}^{(\ell)}} \tag{59}$$

$$= \frac{1}{N^c s(\boldsymbol{\theta})} \sum_{\ell=1}^{L} \sum_{i,j} \sum_{m,n} \frac{\partial h_\mu^{(L)}}{\partial h_{\mu,m}^{(\ell)}} \frac{\partial h_{\mu,m}^{(\ell)}}{\partial W_{ij}^{(\ell)}} \frac{\partial h_\nu^{(L)}}{\partial h_{\nu,n}^{(\ell)}} \frac{\partial h_{\nu,n}^{(\ell)}}{\partial W_{ij}^{(\ell)}} \tag{60}$$

$$= \frac{1}{N^c s(\boldsymbol{\theta})} \left[\frac{1}{N^{2a_L}} \boldsymbol{h}_\mu^{(L-1)} \cdot \boldsymbol{h}_\nu^{(L-1)} + \sum_{\ell=2}^{L-1} \frac{\partial h_\mu^{(L)}}{\partial \boldsymbol{h}_\mu^{(\ell)}} \cdot \frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{h}_\nu^{(\ell)}} \frac{1}{N^{2a_\ell}} \boldsymbol{h}_\mu^{(\ell-1)} \cdot \boldsymbol{h}_\nu^{(\ell-1)} \right. \tag{61}$$

$$\left. + \frac{\partial h_\mu^{(L)}}{\partial \boldsymbol{h}_\mu^{(1)}} \cdot \frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{h}_\nu^{(1)}} \frac{1}{N^{2a_1} D} \boldsymbol{x}_\mu \cdot \boldsymbol{x}_\nu \right] \tag{62}$$

$$= \frac{1}{N^c s(\boldsymbol{\theta})} \left[\frac{1}{N^{2a_L - 1}} \Phi_{\mu\nu}^{(L-1)} + \sum_{\ell=2}^{L-1} \frac{1}{N^{2a_\ell - 1}} G_{\mu\nu}^{(\ell)} \Phi_{\mu\nu}^{(\ell-1)} + \frac{1}{N^{2a_1}} G_{\mu\nu}^{(1)} \Phi_{\mu\nu}^{(0)} \right]. \tag{63}$$

Again, note that this is simply the NTK scaled by the learning rate and the equilibrated energy rescaling. As mentioned in §A.4.2, the feature and gradient kernels concentrate in the limit and are $\Theta_N(1)$. Similarly, we showed that the equilibrated energy rescaling also converges, scaling as $s(\boldsymbol{\theta}) = 1 + \Theta(N^{-2d})$ at constant depth. Using these facts, we get

$$\frac{1}{N^c(1 + N^{-2d})} \left[\frac{1}{N^{2a_L - 1}} \Theta_N(1) + \sum_{\ell=2}^{L-1} \frac{1}{N^{2a_\ell - 1}} \Theta_N(1) \Theta_N(1) + \frac{1}{N^{2a_1}} \Theta_N(1) \Theta_N(1) \right]. \tag{64}$$

We see that the denominator contains two competing scaling factors: N^c and N^{c-2d}. As $N \to \infty$, the behavior of the network is dictated by whichever of these terms is dominant. We require the dominant exponent to be zero, which implies

$$\max(c + 2a_\ell - 1,\ c + 2a_\ell - 1 - 2d) = 0 \implies \boxed{c + 2a_\ell - 1 = \min(0, 2d)}, \tag{65}$$

$$\max(c + 2a_1,\ c + 2a_1 - 2d) = 0 \implies \boxed{c + 2a_1 = \min(0, 2d)}. \tag{66}$$

Note that, similar to the first desideratum (§A.4.1), we find a different constraint for the first layer, again because it is the only layer whose input dimension is not assumed to grow to infinity.

Extra energy-gradient term r. We now examine the second term in Eq. 57, which is specific to PC. For a hidden weight matrix $\boldsymbol{W}^{(\ell)}$, with $\ell \in \{2, \ldots, L-1\}$, we have

$$\frac{\partial f_\mu}{\partial W_{ij}^{(\ell)}} = \Theta_N \left(N^{-d - a_\ell - 1/2} \right), \tag{67}$$

$$\frac{\partial s}{\partial W_{ij}^{(\ell)}} = O \left(\ell N^{-2d - a_\ell - 1} \right). \tag{68}$$

Consequently, the contribution of layer ℓ to r is bounded by

$$r_\ell = O \left(\frac{\gamma^2}{N^c s(\boldsymbol{\theta})^2} \sum_{i,j=1}^{N} N^{-d - a_\ell - 1/2} \ell N^{-2d - a_\ell - 1} \right) \tag{69}$$

$$= O \left(\frac{\ell}{s(\boldsymbol{\theta})^2} N^{1/2 - c - d - 2a_\ell} \right). \tag{70}$$

The output weights give the same width scaling, since

$$\frac{\partial f_\mu}{\partial w_i^{(L)}} = \Theta_N\left(N^{-d-a_L}\right), \qquad \frac{\partial s}{\partial w_i^{(L)}} = O\left(LN^{-2d-a_L-1/2}\right), \tag{71}$$

and the output layer contains N parameters. At fixed depth, using $s(\boldsymbol{\theta}) = 1 + \Theta(N^{-2d})$, we therefore obtain

$$r = O\left(\frac{N^{-(c+d+2a_\ell-1/2)}}{\left(1+N^{-2d}\right)^2}\right). \tag{72}$$

Given the constraint from the rescaled NTK term, $c + 2a_\ell - 1 = 0$, this becomes

$$\boxed{r = O(N^{-1/2-|d|}).} \tag{73}$$

We see that the scaling of the PC-specific energy gradient term r depends on the output scaling exponent d (Eq. 11). In the next section, we will see that the constraint of the last desideratum ($d = 1/2$) forces this term to vanish, while the rescaled NTK term q remains $\Theta_N(1)$.

A.4.4. Desideratum 3: Features evolve in $\Theta_N(1)$ time during training

This final desideratum requires that $dh_i^{(\ell)}/dt = \Theta_N(1)$. Starting again with the first layer, it is useful to write the learning dynamics of the first weight matrix

$$\frac{d\boldsymbol{W}^{(1)}}{dt} = -\eta\frac{\partial \mathcal{F}_\nu^*}{\partial \boldsymbol{W}^{(1)}} = -\eta\frac{1}{s(\boldsymbol{\theta})P}\sum_{\nu=1}^{P}\frac{\partial \mathcal{L}_\nu}{\partial \boldsymbol{W}^{(1)}} \tag{74}$$

$$= \eta\frac{1}{s(\boldsymbol{\theta})P}\sum_{\nu=1}^{P}\Delta_\nu\frac{\partial f_\nu}{\partial \boldsymbol{W}^{(1)}} \tag{75}$$

$$= \frac{\eta_0\gamma^2}{N^c s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\frac{\Delta_\nu}{\gamma}\frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{W}^{(1)}} \tag{76}$$

$$= \frac{\eta_0\gamma}{N^c s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\Delta_\nu\frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{h}_\nu^{(1)}}\frac{\partial \boldsymbol{h}_\nu^{(1)}}{\partial \boldsymbol{W}^{(1)}} \tag{77}$$

$$= \frac{\eta_0\gamma_0}{N^{c+a_1-d}s(\boldsymbol{\theta})}\frac{1}{\sqrt{D}P}\sum_{\nu=1}^{P}\frac{\Delta_\nu\sqrt{N}}{\sqrt{N}}\frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{h}_\nu^{(1)}}\boldsymbol{x}_\nu^\top \tag{78}$$

$$= \frac{\eta_0\gamma_0}{N^{c+a_1-d+1/2}s(\boldsymbol{\theta})}\frac{1}{\sqrt{D}P}\sum_{\nu=1}^{P}\Delta_\nu\boldsymbol{g}_\nu^{(1)}\boldsymbol{x}_\nu^\top, \tag{79}$$

where recall that $\Delta_\nu = -\partial\mathcal{L}_\nu/\partial f_\nu$. Since the equilibrated energy rescaling (Eq. 27) does not depend on the weights of the first layer, we see that their dynamics are simply given by a rescaled loss gradient. Now, let us look at the evolution of the activations of the first layer in terms of their weights

$$\frac{d\boldsymbol{h}_\mu^{(1)}}{dt} = \frac{1}{N^{a_1}}\frac{d\boldsymbol{W}^{(1)}}{dt}\frac{1}{\sqrt{D}}\boldsymbol{x}_\mu \tag{80}$$

$$= \frac{\eta_0\gamma_0}{N^{c+2a_1-d+1/2}s(\boldsymbol{\theta})}\frac{1}{DP}\sum_{\nu=1}^{P}\Delta_\nu\boldsymbol{g}_\nu^{(1)}\boldsymbol{x}_\nu\cdot\boldsymbol{x}_\mu \tag{81}$$

$$= \frac{\eta_0}{N^{c+2a_1-d+1/2}s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\Delta_\nu\boldsymbol{g}_\nu^{(1)}\Phi_{\mu\nu}^{(0)}. \tag{82}$$

Given the results of Eqs. 43-44 & 49, we obtain $c + 2a_1 - d + 1/2 = \min(0, 2d)$. Given the Desideratum 2 constraint $c + 2a_1 = \min(0, 2d)$ (§A.4.3), we get

$$\boxed{d = 1/2}. \tag{83}$$

At $d = 1/2$, the rescaled NTK term q is $\Theta_N(1)$, whereas the PC-specific term r vanishes as $O(N^{-1})$. Thus, the network-function dynamics are asymptotically dominated by the rescaled BP-like term.

Resuming our derivation and moving on to the next layer, we again start by writing the dynamics of the weights

$$\frac{d\boldsymbol{W}^{(\ell)}}{dt} = -\eta \frac{\partial \mathcal{F}^*_\nu}{\partial \boldsymbol{W}^{(\ell)}} = -\eta \frac{1}{s(\boldsymbol{\theta})P} \sum_{\nu=1}^{P} \frac{\partial \mathcal{L}_\nu}{\partial \boldsymbol{W}^{(\ell)}} + \eta \frac{1}{P} \sum_{\nu=1}^{P} \frac{\mathcal{L}_\nu}{s(\boldsymbol{\theta})^2} \frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(\ell)}} \tag{84}$$

$$= \frac{\eta_0 \gamma^2}{N^c} \frac{1}{s(\boldsymbol{\theta})P} \sum_{\nu=1}^{P} \frac{\Delta_\nu}{\gamma} \frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{h}_\nu^{(\ell)}} \frac{\partial \boldsymbol{h}_\nu^{(\ell)}}{\partial \boldsymbol{W}^{(\ell)}} + \frac{\eta_0 \gamma^2}{N^c} \frac{1}{P} \sum_{\nu=1}^{P} \frac{\mathcal{L}_\nu}{s(\boldsymbol{\theta})^2} \frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(\ell)}} \tag{85}$$

$$= \frac{\eta_0 \gamma_0}{N^{c+a_\ell-d+1/2}} \frac{1}{s(\boldsymbol{\theta})P} \sum_{\nu=1}^{P} \Delta_\nu \boldsymbol{g}_\nu^{(\ell)} (\boldsymbol{h}_\nu^{(\ell-1)})^\top + \frac{\eta_0 \gamma^2}{N^c} \frac{1}{P} \sum_{\nu=1}^{P} \frac{\mathcal{L}_\nu}{s(\boldsymbol{\theta})^2} \frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(\ell)}}. \tag{86}$$

Note that now, for $\ell > 1$, we have an extra energy gradient term. To simplify notation, let $\boldsymbol{D}^{(\ell)} := \frac{\eta_0 \gamma^2}{N^c} \frac{1}{P} \sum_{\nu=1}^{P} \frac{\mathcal{L}_\nu}{s(\boldsymbol{\theta})^2} \frac{\partial s}{\partial \boldsymbol{W}^{(\ell)}}$. The evolution of the activations is then

$$\frac{d\boldsymbol{h}_\mu^{(\ell)}}{dt} = \frac{1}{N^{a_\ell}} \frac{d\boldsymbol{W}^{(\ell)}}{dt} \boldsymbol{h}_\mu^{(\ell-1)} + \frac{1}{N^{a_\ell}} \boldsymbol{W}^{(\ell)} \frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt} \tag{87}$$

$$= \frac{\eta_0 \gamma_0}{N^{c+2a_\ell-d+1/2}} \frac{1}{s(\boldsymbol{\theta})P} \sum_{\nu=1}^{P} \Delta_\nu \boldsymbol{g}_\nu^{(\ell)} \frac{N}{N} \boldsymbol{h}_\nu^{(\ell-1)} \cdot \boldsymbol{h}_\nu^{(\ell-1)} + \frac{1}{N^{a_\ell}} \boldsymbol{D}^{(\ell)} \boldsymbol{h}_\mu^{(\ell-1)} + \frac{1}{N^{a_\ell}} \boldsymbol{W}^{(\ell)} \frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt} \tag{88}$$

$$= \frac{\eta_0 \gamma_0}{N^{c+2a_\ell-d-1/2}} \frac{1}{s(\boldsymbol{\theta})P} \sum_{\nu=1}^{P} \Delta_\nu \boldsymbol{g}_\nu^{(\ell)} \Phi_{\mu\nu}^{(\ell-1)} + \frac{1}{N^{a_\ell}} \boldsymbol{D}^{(\ell)} \boldsymbol{h}_\mu^{(\ell-1)} + \frac{1}{N^{a_\ell}} \boldsymbol{W}^{(\ell)} \frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt}. \tag{89}$$

The first term implies that $c + 2a_\ell - d - 1/2 = \min(0, 2d)$ which, under the Desideratum 2 constraint (Eq. 65), again gives

$$\boxed{d = 1/2}. \tag{90}$$

We next verify that the second term in Eq. 89, which arises from the derivative of the equilibrated energy rescaling, is subleading. Using $\partial s / \partial W_{ik}^{(\ell)} = O\left(\ell N^{-2d-a_\ell-1}\right)$ and recalling that $\gamma^2 = \gamma_0^2 N^{2d}$ and $\mathcal{L}_\nu = \Theta_N(1)$, an entry of $\boldsymbol{D}^{(\ell)}$ satisfies

$$D_{ik}^{(\ell)} = O\left(\frac{\gamma^2}{N^c} \frac{\ell N^{-2d-a_\ell-1}}{s(\boldsymbol{\theta})^2}\right) \tag{91}$$

$$= O\left(\frac{\ell N^{-c-a_\ell-1}}{s(\boldsymbol{\theta})^2}\right). \tag{92}$$

Therefore, using $h_{\mu,k}^{(\ell-1)} = \Theta_N(1)$, the corresponding direct contribution to the feature dynamics is bounded by

$$\left[\frac{1}{N^{a_\ell}} \boldsymbol{D}^{(\ell)} \boldsymbol{h}_\mu^{(\ell-1)}\right]_i = \frac{1}{N^{a_\ell}} \sum_{k=1}^{N} D_{ik}^{(\ell)} h_{\mu,k}^{(\ell-1)} \tag{93}$$

$$= O\left(\frac{\ell N^{-c-2a_\ell}}{s(\boldsymbol{\theta})^2}\right) \tag{94}$$

$$= O\left(\frac{\ell N^{-c-2a_\ell}}{(1+N^{-2d})^2}\right). \tag{95}$$

Given the Desideratum 2 constraint $c + 2a_\ell - 1 = 0$, this gives

$$\left[\frac{1}{N^{a_\ell}} \boldsymbol{D}^{(\ell)} \boldsymbol{h}_\mu^{(\ell-1)}\right]_i = O\left(\frac{\ell}{N}\right). \tag{96}$$

At fixed depth, the PC-specific contribution therefore vanishes as $O(N^{-1})$ and is subleading relative to the $\Theta_N(1)$ BP-like direct feature update. Finally, the third term in Eq. 89 propagates feature changes generated in earlier layers. By the forward-pass stability constraint, a $\Theta_N(1)$ change in $\boldsymbol{h}_\mu^{(\ell-1)}$ produces a $\Theta_N(1)$ change in this term. Hence, by induction, the features of every hidden layer evolve on a $\Theta_N(1)$ timescale.

A.4.5. Combining constraints

Putting together the constraints derived from all the desiderata (§A.4.1-A.4.4), we arrive at the following parameterisations:

1. Preactivations are $\Theta_N(1)$ at initialisation: $2a_\ell + b_\ell = 1$ for $\ell \in \{2, \dots, L\}$, and $2a_1 + b_1 = 0$.
2. Networks predictions are $\Theta_N(1)$ during training: $c + 2a_\ell = 1$ for $\ell \in \{2, \dots, L\}$, and $c + 2a_1 = 0$.
3. Features evolve in $\Theta_N(1)$ time during training: $d = 1/2$.

This is the same set of one-dimensional solutions derived for BP, thus concluding the proof of Theorem 1. See Figure A.5 for a visual illustration.

A.5. Derivation of depth scalings for linear residual PCNs

Following the derivation of the width scalings in §A.4, this section derives the depth scaling exponent α that satisfies the 2 Desiderata described in §4, proving Theorem 2. Following previous work (Yang et al., 2023; Bordelon et al., 2023), we fix the width parameterisation to the mean-field (see Table 2) and extend it to residual networks as follows:[5]

$$f_\mu = \frac{1}{\gamma_0 \sqrt{N}} h_\mu^{(L)}, \tag{97}$$

$$h_\mu^{(L)} = \frac{1}{\sqrt{N}} \boldsymbol{w}^{(L)} \boldsymbol{h}_\mu^{(L-1)}, \tag{98}$$

$$\boldsymbol{h}_\mu^{(\ell)} = \left(\boldsymbol{I} + \frac{1}{L^\alpha \sqrt{N}} \boldsymbol{W}^{(\ell)} \right) \boldsymbol{h}_\mu^{(\ell-1)}, \tag{99}$$

$$\boldsymbol{h}_\mu^{(1)} = \frac{1}{\sqrt{D}} \boldsymbol{W}^{(1)} \boldsymbol{x}_\mu, \tag{100}$$

where α is some depth-dependent exponent scaling the residual branches. Recall that under our (mean-field) parameterisation, all the weights are initialised as $\theta_i \sim \mathcal{N}(0, 1)$, and $\gamma = \gamma_0 \sqrt{N}$ since $d = 1/2$. We will again consider gradient flow on the equilibrated energy (Eq. 5), with learning rate $\eta = \eta_0 \gamma^2 N^{-c} = \eta_0 \gamma_0^2 N$ since $c = 0$.

A.5.1. Desideratum 4: Preactivations are $\Theta_L(1)$ at initialisation

Similar to the first desideratum (§A.4.1), we require that the mean-squared values of the residual stream variables (Eq. 19) are $\Theta_L(1)$ at initialisation. Similar derivations have been performed in previous work (Hayou & Yang, 2023; Hayou et al., 2021), but we again include example calculations for completeness.

We showed in §A.4.1 that the mean of the preactivations at each layer is zero (because of the zero-mean weight initialisation), and this fact does not change for residual networks. The covariance satisfies the following recurrence relation:

$$\left\langle h_{\mu,i}^{(\ell)} h_{\nu,j}^{(\ell)} \right\rangle = \left\langle h_{\mu,i}^{(\ell-1)} h_{\nu,j}^{(\ell-1)} \right\rangle + \frac{1}{L^{2\alpha} N} \sum_{k=1}^{N} \sum_{k'=1}^{N} \left\langle W_{ik}^{(\ell)}(0) W_{jk'}^{(\ell)}(0) \right\rangle \left\langle h_{\mu,k}^{(\ell-1)} h_{\nu,k'}^{(\ell-1)} \right\rangle \tag{101}$$

$$= \left\langle h_{\mu,i}^{(\ell-1)} h_{\nu,j}^{(\ell-1)} \right\rangle + \delta_{ij} \frac{1}{L^{2\alpha}} \left\langle h_{\mu,1}^{(\ell-1)} h_{\nu,1}^{(\ell-1)} \right\rangle \tag{102}$$

$$= \delta_{ij} \left\langle h_{\mu,1}^{(\ell-1)} h_{\nu,1}^{(\ell-1)} \right\rangle \left(1 + \frac{1}{L^{2\alpha}} \right), \tag{103}$$

where, as in §A.4.1, we used the fact that at initialization the neurons are i.i.d., so $\langle h_{\mu,i}^{(\ell-1)} h_{\nu,j}^{(\ell-1)} \rangle = \delta_{ij} \langle h_{\mu,1}^{(\ell-1)} h_{\nu,1}^{(\ell-1)} \rangle$. Unrolling this recurrence up to the last hidden activation $h_{\mu,i}^{(L-1)}$, we get

$$\left\langle h_{\mu,i}^{(L-1)} h_{\nu,j}^{(L-1)} \right\rangle = \delta_{ij} \Phi_{\mu\nu}^{(0)} \left(1 + \frac{1}{L^{2\alpha}} \right)^{L-2}, \tag{104}$$

[5] Note that the specific choice of width-stable and feature-learning parameterisation (Eq. 15) does not affect the results of the depth scaling analysis, since they all satisfy the same desiderata.

where recall from Eq. 40 that $\Phi^{(0)}_{\mu\nu}$ is the normalised feature kernel of the data. To keep this $\Theta_L(1)$, we need

$$\boxed{\alpha \geq 1/2}. \tag{105}$$

Note that for $\alpha = 1/2$ each residual branch has exactly a $\Theta(1/L)$ contribution to the variance, while for $\alpha > 1/2$ their contribution vanishes, and the covariance reduces to the data kernel. In contrast, plain MLPs do not have a depth-dependent residual scaling. Stable signal propagation instead requires the layer Jacobians (i.e. derivatives of the network output with respect to a layer's activations) to remain close to isometric (with singular values close to 1), and at large depth this can in practice only be achieved with orthogonal initialisation and suitable activation functions (Poole et al., 2016; Schoenholz et al., 2016).

A.5.2. Useful asymptotic results for residual networks

Before moving on to the next desideratum, it is useful to extend some of the asymptotic results derived for MLPs in §A.4.2 to residual networks. In particular, we will extend the PC-specific results about the equilibrated energy rescaling and its parameter derivative.

Equilibrated energy rescaling. First, under our considered residual network parameterisation (Eqs. 97-100), the equilibrated energy rescaling takes the following form (Innocenti et al., 2025):

$$s(\boldsymbol{\theta}) = 1 + \frac{1}{\gamma_0^2 N^2}\left(\|\boldsymbol{w}^{(L)}\|^2 + \sum_{\ell=2}^{L-1}\left\|\boldsymbol{p}^{(L:\ell)}\right\|^2\right), \tag{106}$$

$$\boldsymbol{p}^{(L:\ell)} \coloneqq \boldsymbol{w}^{(L)}\prod_{j=\ell}^{L-1}\left(\boldsymbol{I}_N + \frac{1}{L^\alpha\sqrt{N}}\boldsymbol{W}^{(j)}\right), \tag{107}$$

where $\boldsymbol{p}^{(L:\ell)}$ denotes the ℓth residual path from $\ell > 1$ to L, and we define the matrix product as $\prod_{\ell=1}^{L}\boldsymbol{A}^{(\ell)} = \boldsymbol{A}^{(L)}\boldsymbol{A}^{(L-1)}\cdots\boldsymbol{A}^{(1)}$. The squared norm of the output weights scales as $\langle\|\boldsymbol{w}^{(L)}\|^2\rangle = \Theta(N)$, and from the previous forward pass constraint $\alpha \geq 1/2$ (§A.5.1), the sum of the squared residual norms is $\langle\sum_{\ell=2}^{L-1}\left\|\boldsymbol{p}^{(L:\ell)}\right\|^2\rangle = \sum_{\ell=2}^{L-1}\Theta(N) = \Theta(LN)$. We therefore get

$$s(\boldsymbol{\theta}) = 1 + \frac{1}{\gamma_0^2 N^2}\left[\Theta(N) + \Theta(LN)\right] \tag{108}$$

$$= 1 + \Theta\left(\frac{1}{N} + \frac{L}{N}\right) = 1 + \Theta\left(L/N\right), \tag{109}$$

which we see asymptotically scales in the same way as for MLPs (Eq. 49).

Rescaling derivative with respect to hidden-layer weights. We will also need the derivative of the equilibrated energy rescaling with respect to a hidden residual weight matrix. Define

$$\boldsymbol{A}^{(j)} \coloneqq \boldsymbol{I} + \frac{1}{L^\alpha\sqrt{N}}\boldsymbol{W}^{(j)}, \tag{110}$$

$$\boldsymbol{P}^{(\ell:k)} \coloneqq \boldsymbol{A}^{(\ell)}\boldsymbol{A}^{(\ell-1)}\cdots\boldsymbol{A}^{(k)}, \qquad k \leq \ell, \tag{111}$$

and set $\boldsymbol{P}^{(\ell:k)} \coloneqq \boldsymbol{I},\ \ell < k$. The residual path from layer k to the output can then be written as $\boldsymbol{p}^{(L:k)} = \boldsymbol{w}^{(L)}\boldsymbol{P}^{(L-1:k)}$. Recall that the equilibrated energy rescaling is

$$s(\boldsymbol{\theta}) = 1 + \frac{1}{\gamma_0^2 N^2}\left(\left\|\boldsymbol{w}^{(L)}\right\|_2^2 + \sum_{k=2}^{L-1}\left\|\boldsymbol{p}^{(L:k)}\right\|_2^2\right). \tag{112}$$

For a fixed hidden layer $\ell \in \{2,\ldots,L-1\}$, only paths beginning at $k \leq \ell$ depend on $\boldsymbol{W}^{(\ell)}$. Differentiating these terms gives

$$\frac{\partial s}{\partial \boldsymbol{W}^{(\ell)}} = \frac{2}{\gamma_0^2 N^2 L^\alpha\sqrt{N}}\sum_{k=2}^{\ell}\left(\boldsymbol{w}^{(L)}\boldsymbol{P}^{(L-1:\ell+1)}\right)^\top\left(\boldsymbol{w}^{(L)}\boldsymbol{P}^{(L-1:k)}\right)\left(\boldsymbol{P}^{(\ell-1:k)}\right)^\top. \tag{113}$$

To determine its scaling, define $\boldsymbol{u}^{(\ell)} := \boldsymbol{w}^{(L)}\boldsymbol{P}^{(L-1:\ell+1)}$ and $\boldsymbol{v}^{(\ell,k)} := \boldsymbol{w}^{(L)}\boldsymbol{P}^{(L-1:k)}\left(\boldsymbol{P}^{(\ell-1:k)}\right)^\top$. Each summand in Eq. (113) is therefore the rank-one matrix $\left(\boldsymbol{u}^{(\ell)}\right)^\top \boldsymbol{v}^{(\ell,k)}$. Under the forward path-stability condition $\alpha \geq 1/2$, the coordinates of $\boldsymbol{u}^{(\ell)}$ and $\boldsymbol{v}^{(\ell,k)}$ remain $O_{N,L}(1)$. Hence, each entry of an individual summand is $O_{N,L}(1)$, and

$$\frac{\partial s}{\partial W_{ij}^{(\ell)}} = O\left(\frac{\ell}{L^\alpha N^{5/2}}\right). \tag{114}$$

In particular, for an interior layer $\ell = \Theta(L)$,

$$\frac{\partial s}{\partial W_{ij}^{(\ell)}} = O\left(L^{1-\alpha}N^{-5/2}\right), \qquad N, L \to \infty. \tag{115}$$

A.5.3. Desideratum 5: Each residual block has a $\Theta(1/L)$ contribution to feature evolution

We also require that the update induced by each residual block on the hidden features is $\Theta(1/L)$, so that the cumulative feature evolution across all L residual blocks remains $\Theta_L(1)$. Recall that under our parameterisation $\gamma = \gamma_0\sqrt{N}$ and $\eta = \eta_0\gamma_0^2 N$. As before, it is useful to first write the dynamics of the weights

$$\frac{d\boldsymbol{W}^{(\ell)}}{dt} = -\eta\frac{\partial \mathcal{F}^*}{\partial \boldsymbol{W}^{(\ell)}} = \frac{\eta_0\gamma_0^2 N}{s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\frac{\Delta_\nu}{\gamma_0\sqrt{N}}\frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{h}_\nu^{(\ell)}}\frac{\partial \boldsymbol{h}_\nu^{(\ell)}}{\partial \boldsymbol{W}^{(\ell)}} + \frac{\eta_0\gamma_0^2 N}{s(\boldsymbol{\theta})^2}\frac{1}{P}\sum_{\nu=1}^{P}\mathcal{L}_\nu\frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(\ell)}} \tag{116}$$

$$= \frac{\eta_0\gamma_0}{N^{-1/2}s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\Delta_\nu\frac{\sqrt{N}}{\sqrt{N}}\frac{\partial h_\nu^{(L)}}{\partial \boldsymbol{h}_\nu^{(\ell)}}\frac{1}{L^\alpha\sqrt{N}}(\boldsymbol{h}_\nu^{(\ell-1)})^\top + \frac{\eta_0\gamma_0^2 N}{s(\boldsymbol{\theta})^2}\frac{1}{P}\sum_{\nu=1}^{P}\mathcal{L}_\nu\frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(\ell)}} \tag{117}$$

$$= \frac{\eta_0\gamma_0}{\sqrt{N}s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\Delta_\nu\boldsymbol{g}_\nu^{(\ell)}\frac{1}{L^\alpha}(\boldsymbol{h}_\nu^{(\ell-1)})^\top + \frac{\eta_0\gamma_0^2 N}{s(\boldsymbol{\theta})^2}\frac{1}{P}\sum_{\nu=1}^{P}\mathcal{L}_\nu\frac{\partial s(\boldsymbol{\theta})}{\partial \boldsymbol{W}^{(\ell)}}. \tag{118}$$

As in §A.4.4, let $\boldsymbol{D}^{(\ell)} := \frac{\eta_0\gamma_0^2 N}{s(\boldsymbol{\theta})^2}\frac{1}{P}\sum_{\nu=1}^{P}\mathcal{L}_\nu\frac{\partial s}{\partial \boldsymbol{W}^{(\ell)}}$ to keep the notation compact. The evolution of any hidden activation for $\ell \in \{2, \dots, L-1\}$ is then

$$\frac{d\boldsymbol{h}_\mu^{(\ell)}}{dt} = \frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt} + \frac{1}{L^\alpha\sqrt{N}}\left(\frac{d\boldsymbol{W}^{(\ell)}}{dt}\boldsymbol{h}_\mu^{(\ell-1)} + \boldsymbol{W}^{(\ell)}\frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt}\right) \tag{119}$$

$$= \frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt} + \frac{1}{L^\alpha\sqrt{N}}\left[\left(\frac{\eta_0\gamma_0}{\sqrt{N}s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\Delta_\nu\boldsymbol{g}_\nu^{(\ell)}\frac{1}{L^\alpha}(\boldsymbol{h}_\nu^{(\ell-1)})^\top + \boldsymbol{D}^{(\ell)}\right)\boldsymbol{h}_\mu^{(\ell-1)} + \boldsymbol{W}^{(\ell)}\frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt}\right] \tag{120}$$

$$= \frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt} + \frac{1}{L^{2\alpha}}\frac{\eta_0\gamma_0}{s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\Delta_\nu\boldsymbol{g}_\nu^{(\ell)}\Phi_{\mu\nu}^{(\ell-1)} + \frac{1}{L^\alpha\sqrt{N}}\left(\boldsymbol{D}^{(\ell)}\boldsymbol{h}_\mu^{(\ell-1)} + \boldsymbol{W}^{(\ell)}\frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt}\right) \tag{121}$$

$$= \frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt} + \frac{1}{L^{2\alpha}}\frac{\eta_0\gamma_0}{s(\boldsymbol{\theta})}\frac{1}{P}\sum_{\nu=1}^{P}\Delta_\nu\boldsymbol{g}_\nu^{(\ell)}\Phi_{\mu\nu}^{(\ell-1)} \tag{122}$$

$$+ \frac{\eta_0\gamma_0^2\sqrt{N}}{L^\alpha s(\boldsymbol{\theta})^2}\frac{1}{P}\sum_{\nu=1}^{P}\mathcal{L}_\nu\frac{\partial s}{\partial \boldsymbol{W}^{(\ell)}}\boldsymbol{h}_\mu^{(\ell-1)} + \frac{1}{L^\alpha\sqrt{N}}\boldsymbol{W}^{(\ell)}\frac{d\boldsymbol{h}_\mu^{(\ell-1)}}{dt}. \tag{123}$$

For an interior residual layer $\ell = \Theta(L)$, the direct BP-like contribution scales as $\Theta\left(\frac{L^{-2\alpha}}{s(\boldsymbol{\theta})}\right)$. Moreover, using Eq. (115),

$$\left[\frac{\partial s}{\partial \boldsymbol{W}^{(\ell)}}\boldsymbol{h}_\mu^{(\ell-1)}\right]_i = O\left(L^{1-\alpha}N^{-3/2}\right), \tag{124}$$

and hence the PC-specific contribution satisfies

$$\frac{\eta_0\gamma_0^2\sqrt{N}}{L^\alpha s(\boldsymbol{\theta})^2}\frac{1}{P}\sum_{\nu=1}^{P}\mathcal{L}_\nu\left[\frac{\partial s}{\partial \boldsymbol{W}^{(\ell)}}\boldsymbol{h}_\mu^{(\ell-1)}\right]_i = O\left(\frac{L^{1-2\alpha}}{Ns(\boldsymbol{\theta})^2}\right). \tag{125}$$

Therefore, coordinatewise,

$$\frac{dh_{\mu,i}^{(\ell)}}{dt} = \frac{dh_{\mu,i}^{(\ell-1)}}{dt} + \Theta\left(\frac{L^{-2\alpha}}{s(\boldsymbol{\theta})}\right) + O\left(\frac{L^{1-2\alpha}}{Ns(\boldsymbol{\theta})^2}\right) + O(L^{-\alpha})\frac{dh_{\mu,i}^{(\ell-1)}}{dt}. \tag{126}$$

Recall from Eq. (21) that $s(\boldsymbol{\theta}) = 1 + \Theta(L/N)$. We therefore consider joint width-depth limits satisfying $L/N = O(1)$, for which $s(\boldsymbol{\theta}) = \Theta(1)$. In this regime, the direct BP-like contribution of one residual block is $\Theta(L^{-2\alpha})$, while its PC-specific correction is

$$O\left(\frac{L}{N}L^{-2\alpha}\right). \tag{127}$$

Thus, the PC-specific term is at most of the same depth order as the BP-like term; in the wider-than-deep regime $L/N \to 0$, it is asymptotically subleading. Assuming the usual non-degeneracy of the leading direct feature update, Desideratum 5 requires $L^{-2\alpha} = \Theta(L^{-1})$, which uniquely gives

$$\boxed{\alpha = 1/2}. \tag{128}$$

At this critical scaling, each residual block has a $\Theta(1/L)$ direct contribution to feature evolution, and the cumulative contribution over L blocks is $\Theta_L(1)$. This completes the proof of Theorem 2 in joint limits satisfying $L/N = O(1)$.

A.6. Convergence of the equilibrated energy gradients to the MSE gradients

Here we derive Corollaries 3.2 & 4.2. Recall the equilibrated PC energy from Eq. 5:

$$\mathcal{F}^*(\boldsymbol{\theta}) = \frac{\mathcal{L}(\boldsymbol{\theta})}{s(\boldsymbol{\theta})}. \tag{129}$$

Differentiating with respect to the parameters of any layer $\boldsymbol{\theta}^{(\ell)}$ gives

$$\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{F}^* = \frac{1}{s(\boldsymbol{\theta})}\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{L} - \frac{\mathcal{L}(\boldsymbol{\theta})}{s(\boldsymbol{\theta})^2}\nabla_{\boldsymbol{\theta}^{(\ell)}}s. \tag{130}$$

Subtracting the MSE gradient, we obtain

$$\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{F}^* - \nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{L} = \left(\frac{1}{s(\boldsymbol{\theta})} - 1\right)\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{L} - \frac{\mathcal{L}(\boldsymbol{\theta})}{s(\boldsymbol{\theta})^2}\nabla_{\boldsymbol{\theta}^{(\ell)}}s. \tag{131}$$

The first term vanishes whenever $s(\boldsymbol{\theta}) \to 1$, while the second term depends on the parameter derivative of the rescaling. Taking norms in Eq. 131 gives

$$\frac{\|\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{F}^* - \nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{L}\|}{\|\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{L}\|} \leq \left|\frac{1}{s(\boldsymbol{\theta})} - 1\right| + \frac{|\mathcal{L}(\boldsymbol{\theta})|}{s(\boldsymbol{\theta})^2}\frac{\|\nabla_{\boldsymbol{\theta}^{(\ell)}}s\|}{\|\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{L}\|}. \tag{132}$$

Therefore, assuming $\mathcal{L}(\boldsymbol{\theta}) = O(1)$, $s(\boldsymbol{\theta}) = \Theta(1)$ and non-vanishing MSE gradients, we get the following two sufficient conditions for the PC gradients to converge to the BP gradients:

$$s(\boldsymbol{\theta}) - 1 = o(1), \qquad \frac{\|\nabla_{\boldsymbol{\theta}^{(\ell)}}s\|}{\|\nabla_{\boldsymbol{\theta}^{(\ell)}}\mathcal{L}\|} = o(1) \quad \text{for every parameter layer } \ell. \tag{133}$$

Linear MLPs. For MLPs, we keep the depth L fixed and take $N \to \infty$, since signal propagation in plain MLPs can become unstable with depth, as discussed in §A.5.1. Under a width-stable and feature-learning parameterisation ($d = 1/2$), the equilibrated energy rescaling scales as $s(\boldsymbol{\theta}) - 1 = O\left(N^{-1}\right)$. For a hidden weight matrix $\boldsymbol{W}^{(\ell)}$ with $\ell \in \{2, \ldots, L-1\}$ the MSE gradient has entrywise scaling

$$\frac{\partial\mathcal{L}}{\partial W_{ij}^{(\ell)}} = \Theta\left(N^{-d-a_\ell-1/2}\right) = \Theta\left(N^{-a_\ell-1}\right), \tag{134}$$

whereas the derivative of the rescaling gives

$$\frac{\partial s}{\partial W_{ij}^{(\ell)}} = O\left(N^{-2d-a_\ell-1}\right) = O\left(N^{-a_\ell-2}\right). \tag{135}$$

Consequently,

$$\frac{\|\nabla_{\boldsymbol{W}^{(\ell)}} s\|_{\mathrm{F}}}{\|\nabla_{\boldsymbol{W}^{(\ell)}} \mathcal{L}\|_{\mathrm{F}}} = O\left(N^{-1}\right). \tag{136}$$

For the output weights,

$$\frac{\partial \mathcal{L}}{\partial w_i^{(L)}} = \Theta\left(N^{-d-a_L}\right) = \Theta\left(N^{-a_L-1/2}\right), \tag{137}$$

$$\frac{\partial s}{\partial w_i^{(L)}} = O\left(N^{-2d-a_L-1/2}\right) = O\left(N^{-a_L-3/2}\right), \tag{138}$$

and hence

$$\frac{\|\nabla_{\boldsymbol{w}^{(L)}} s\|}{\|\nabla_{\boldsymbol{w}^{(L)}} \mathcal{L}\|} = O\left(N^{-1}\right). \tag{139}$$

The rescaling does not depend on $\boldsymbol{W}^{(1)}$, so its derivative is zero for the first layer. Combining these results, we obtain

$$\frac{\|\nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{F}^* - \nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L}\|}{\|\nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L}\|} = O\left(N^{-1}\right) \tag{140}$$

for the weights of every layer ℓ whose MSE gradient remains non-vanishing. Thus, at fixed depth L, the equilibrated energy gradients converge to the BP MSE gradients at rate $O(N^{-1})$ as $N \to \infty$.

Linear residual networks. It is relatively straightforward to generalise this result to linear residual networks under width- and depth-stable and feature-learning parameterisations (Eqs. 97–100), with the critical depth exponent $\alpha = 1/2$. Unlike the MLP result above, here we consider joint width-depth limits $N, L \to \infty$. The exponent $\alpha = 1/2$ ensures depth-stable signal propagation, allowing the following estimates to hold as both N and L grow. The rescaling satisfies $s(\boldsymbol{\theta}) - 1 = O\left(L/N\right)$.

For a hidden residual matrix, we now have the following scalings

$$\frac{\partial \mathcal{L}}{\partial W_{ij}^{(\ell)}} = \Theta\left(L^{-\alpha} N^{-3/2}\right), \tag{141}$$

$$\frac{\partial s}{\partial W_{ij}^{(\ell)}} = O\left(\ell L^{-\alpha} N^{-5/2}\right) = O\left(L^{1-\alpha} N^{-5/2}\right). \tag{142}$$

Hence, $\frac{\|\nabla_{\boldsymbol{W}^{(\ell)}} s\|_{\mathrm{F}}}{\|\nabla_{\boldsymbol{W}^{(\ell)}} \mathcal{L}\|_{\mathrm{F}}} = O\left(L/N\right)$. Similarly, for the output weights,

$$\frac{\partial \mathcal{L}}{\partial w_i^{(L)}} = \Theta(N^{-1}), \qquad \frac{\partial s}{\partial w_i^{(L)}} = O(LN^{-2}), \tag{143}$$

and therefore $\frac{\|\nabla_{\boldsymbol{w}^{(L)}} s\|}{\|\nabla_{\boldsymbol{w}^{(L)}} \mathcal{L}\|} = O\left(L/N\right)$. Substituting into Eq. 132 again gives

$$\frac{\|\nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{F}^* - \nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L}\|}{\|\nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L}\|} = O\left(\frac{L}{N}\right). \tag{144}$$

Hence, in any joint width-depth limit satisfying $L/N \to 0$, both terms in Eq. 131 vanish relative to the MSE gradient, and the equilibrated energy gradients converge layerwise to the BP MSE gradients.

Convergence of the complete gradient vector. The previous results are stated layerwise, but they also imply convergence of the complete gradient vector. Let $\boldsymbol{\theta}$ denote the concatenation of all vectorised parameter blocks, and define

$$\boldsymbol{\delta}^{(\ell)} \coloneqq \nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{F}^* - \nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L}. \tag{145}$$

Suppose that, uniformly over the parameter layers,

$$\left\| \boldsymbol{\delta}^{(\ell)} \right\|_2 \leq C_{N,L} \left\| \nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L} \right\|_2 , \tag{146}$$

where $C_{N,L} = O(N^{-1})$ for MLPs at fixed depth and $C_{N,L} = O(L/N)$ for residual networks. Since the complete gradient is obtained by concatenating the layerwise gradient blocks,

$$\begin{aligned} \left\| \nabla_{\boldsymbol{\theta}} \mathcal{F}^* - \nabla_{\boldsymbol{\theta}} \mathcal{L} \right\|_2^2 &= \sum_{\ell=1}^{L} \left\| \boldsymbol{\delta}^{(\ell)} \right\|_2^2 && (147) \\ &\leq C_{N,L}^2 \sum_{\ell=1}^{L} \left\| \nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L} \right\|_2^2 && (148) \\ &= C_{N,L}^2 \left\| \nabla_{\boldsymbol{\theta}} \mathcal{L} \right\|_2^2 . && (149) \end{aligned}$$

Therefore,

$$\frac{\left\| \nabla_{\boldsymbol{\theta}} \mathcal{F}^* - \nabla_{\boldsymbol{\theta}} \mathcal{L} \right\|_2}{\left\| \nabla_{\boldsymbol{\theta}} \mathcal{L} \right\|_2} \leq C_{N,L}. \tag{150}$$

Consequently, provided that the complete MSE gradient remains non-vanishing, the complete equilibrated energy gradient converges to the complete BP gradient at rate $O(N^{-1})$ for MLPs at fixed depth, and at rate $O(L/N)$ for residual networks.

A.7. Heuristic extension to other architectures and optimisers

Because we showed that the set of stable and feature-learning parameterisations for linear PCNs is the same as for BP (Theorems 1-2), we can heuristically extend our results to closely related architectures such as CNNs and other optimisers such as Adam (Kingma & Ba, 2014). For Adam, since it is approximately scale-invariant to the gradient, we would also need to further scale the learning rate (of hidden matrices) η by $N^{-1/2} L^{\alpha-1}$ to keep the feature updates stable with the width and depth (Yang & Littwin, 2023; Bordelon et al., 2024; Dey et al., 2025).

A.7.1. CNNs

For CNNs, we can adjust the scalings (Table 1) by essentially taking the number of channels as the width. The mean-field-parameterised PC energy for the CNN architecture presented in Figure 4 is as follows:

$$\begin{aligned} \mathcal{F}(\boldsymbol{Z}, \boldsymbol{\theta}) = \frac{1}{2P} \sum_{\mu=1}^{P} \Big(& \\ & \| \boldsymbol{Z}_\mu^{(1)} - \underbrace{\frac{1}{\sqrt{D}} \boldsymbol{W}^{(1)} \boldsymbol{X}_\mu}_{\text{2D Conv.}} \|_F^2 + \| \boldsymbol{Z}_\mu^{(2)} - \underbrace{(\frac{1}{\sqrt{NL}} \boldsymbol{W}^{(2)} \phi(\boldsymbol{Z}_\mu^{(1)}) + \boldsymbol{Z}_\mu^{(1)})}_{\text{Res. block}} \|_F^2 + \| \boldsymbol{Z}_\mu^{(3)} - \underbrace{\boldsymbol{W}^{(3)} \boldsymbol{Z}_\mu^{(2)}}_{\text{Avg. pool.}} \|_F^2 \quad \text{[Stage 1]} \\ & + \| \boldsymbol{Z}_\mu^{(4)} - \frac{1}{\sqrt{N}} \boldsymbol{W}^{(4)} \boldsymbol{Z}_\mu^{(3)} \|_F^2 + \| \boldsymbol{Z}_\mu^{(5)} - (\frac{1}{\sqrt{NL}} \boldsymbol{W}^{(5)} \phi(\boldsymbol{Z}_\mu^{(4)}) + \boldsymbol{Z}_\mu^{(4)}) \|_F^2 + \| \boldsymbol{Z}_\mu^{(6)} - \boldsymbol{W}^{(6)} \boldsymbol{Z}_\mu^{(5)} \|_F^2 \quad \text{[Stage 2]} \\ & + \| \boldsymbol{Z}_\mu^{(7)} - \frac{1}{\sqrt{N}} \boldsymbol{W}^{(7)} \boldsymbol{Z}_\mu^{(6)} \|_F^2 + \| \boldsymbol{Z}_\mu^{(8)} - (\frac{1}{\sqrt{NL}} \boldsymbol{W}^{(8)} \phi(\boldsymbol{Z}_\mu^{(7)}) + \boldsymbol{Z}_\mu^{(7)}) \|_F^2 + \| \boldsymbol{Z}_\mu^{(9)} - \boldsymbol{W}^{(9)} \boldsymbol{Z}_\mu^{(8)} \|_F^2 \quad \text{[Stage 3]} \\ & + \mathcal{L}_{\text{CE}} \big(\underbrace{\frac{1}{C_{\text{out}} H' W'} \boldsymbol{W}^{(10)} \text{vec}(\boldsymbol{Z}_\mu^{(9)})}_{\text{Readout}}, \boldsymbol{y}_\mu \big) \\ \Big) . & \end{aligned} \tag{151}$$

This architecture can be described as a "3-stage ResNet", where each stage applies:

1. a 2D convolution $\boldsymbol{W}^{(\ell)} \in \mathbb{R}^{C_{\text{out}} \times (k^2 C_{\text{in}})}$ with kernel size k and input and output channels $(C_{\text{in}}, C_{\text{out}})$;
2. one residual block consisting of a nonlinearity $\phi(\cdot)$ plus convolution; and
3. an average pooling operation expressed as a fixed sparse linear transformation $\boldsymbol{W}^{(\ell)}$ for $\ell \in \{3, 6, 9\}$, with kernel size $k = 2$ and stride $s = 2$,

before a final fully connected layer $\boldsymbol{W}^{(10)} \in \mathbb{R}^{C \times (C_{\text{out}} H' W')}$, where C is the number of classes, and H' and W' are the input's final height and width dimensions, respectively. For the experiment in Figure 4, we used Tanh as nonlinearity. As before, all the weights are initialised from a standard Gaussian, $W_{ij}^{(\ell)} \sim \mathcal{N}(0, 1)$. The last term of Eq. 151 is the cross-entropy loss, i.e. $\mathcal{L}_{\text{CE}}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_{c=1}^{C} y_c [\log \text{softmax}(\hat{\boldsymbol{y}})]_c$. All convolutions have unit stride and padding. The input image $\boldsymbol{X}_\mu$ is transformed to have size $(k^2 C_{\text{in}}) \times HW$, so that 2D convolutions can be expressed as simple matrix multiplications. Note that here $D = k^2 C_{\text{in}}$. As in the experiment in Figure 4, both the kernel size and the number of output channels are kept constant across all convolutions, giving an effective model width of $N = k^2 C_{\text{out}}$.[6] The depth scaling factor L is given by the total number of residual blocks (which is what is theoretically taken to infinity), in this case 3 (with one block in each stage). The experiments in Figure 4 add non-residual layers to the depth scaling, following Bordelon et al. (2023).

A.7.2. Transformers

Let $\boldsymbol{X}_\mu \in \{1, \dots, V\}^T$ denote a sequence of input-token indices of length T, and let $\boldsymbol{y}_\mu \in \{1, \dots, V\}^T$ denote the corresponding next-token targets. The transformer has width N, L residual blocks, and activities $\boldsymbol{Z}_\mu^{(\ell)} \in \mathbb{R}^{T \times N}$. Define the embedding and transformer-blocks as

$$\text{Emb}(\boldsymbol{X}) \coloneqq \boldsymbol{W}^{(\text{tok.})}[\boldsymbol{X}] + \boldsymbol{W}^{(\text{pos.})}, \tag{152}$$

$$\text{Block}(\boldsymbol{Z}) \coloneqq \boldsymbol{Z} + \frac{1}{L}\text{MHA}(\boldsymbol{Z}) + \frac{1}{L}\text{MLP}\left(\boldsymbol{Z} + \frac{1}{L}\text{MHA}(\boldsymbol{Z})\right). \tag{153}$$

where $\boldsymbol{W}^{(\text{tok.})}[\boldsymbol{X}] \in \mathbb{R}^{V \times N}$ denotes row-wise embedding lookup, and $\boldsymbol{W}^{(\text{pos.})} \in \mathbb{R}^{T \times N}$ is a positional embedding matrix. The PC energy used for the transformer experiment in Figure 4 is

$$\begin{aligned}\mathcal{F}(\boldsymbol{Z}, \boldsymbol{\theta}) = \frac{1}{2TP} \sum_{\mu=1}^{P} \left[\left\| \boldsymbol{Z}_\mu^{(1)} - \text{Emb}(\boldsymbol{X}_\mu) \right\|_F^2 + \sum_{\ell=2}^{L+1} \left\| \boldsymbol{Z}_\mu^{(\ell)} - \text{Block}\left(\boldsymbol{Z}_\mu^{(\ell-1)}\right) \right\|_F^2 \right] \\ + \frac{1}{P} \sum_{\mu=1}^{P} \mathcal{L}_{\text{CE}}\left(\frac{1}{N} \boldsymbol{Z}_\mu^{(L+1)} \boldsymbol{W}^{(\text{out})}, \boldsymbol{y}_\mu \right),\end{aligned} \tag{154}$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss averaged over sequence positions, and $\boldsymbol{W}^{(\text{out})} \in \mathbb{R}^{N \times V}$. For h attention heads with dimension $d = N/h$, multi-head self-attention (MHA) (Vaswani et al., 2017) is

$$\text{MHA}(\boldsymbol{Z}) = \text{concat}\left(\text{head}_1(\boldsymbol{Z}), \dots, \text{head}_h(\boldsymbol{Z})\right) \frac{1}{\sqrt{N}} \boldsymbol{W}^{(O)}, \tag{155}$$

$$\text{head}_i(\boldsymbol{Z}) = \text{softmax}\left(\frac{(\boldsymbol{Z}\boldsymbol{W}^{(Q,i)})(\boldsymbol{Z}\boldsymbol{W}^{(K,i)})^\top}{d} \right) \boldsymbol{Z}\boldsymbol{W}^{(V,i)}, \tag{156}$$

where the softmax is applied row-wise, $\boldsymbol{W}^{(Q,i)}, \boldsymbol{W}^{(K,i)}, \boldsymbol{W}^{(V,i)} \in \mathbb{R}^{N \times d}$, $\boldsymbol{W}^{(O)} \in \mathbb{R}^{N \times N}$, and we omit the causal mask for simplicity. The feedforward subblock is

$$\text{MLP}(\boldsymbol{Z}) = \phi\left(\frac{1}{\sqrt{4N}} \boldsymbol{Z}\boldsymbol{W}^{(1)} \right) \frac{1}{\sqrt{N}} \boldsymbol{W}^{(2)}, \tag{157}$$

where following standard practice the first MLP layer expands the model width by a factor of 4, with $\boldsymbol{W}^{(1)} \in \mathbb{R}^{N \times 4N}$ and $\boldsymbol{W}^{(2)} \in \mathbb{R}^{4N \times N}$. For the experiment in Figure 4, we used $L = 12$ blocks, $h = 8$ attention heads, and GeLU as nonlinearity. Layer normalisation was not used. All the weights were again initialised from a standard Gaussian. The overall parameterisation is adapted from Dey et al. (2025), and we used their per-parameter Adam learning-rate scalings.

[6] Note that, even though it is the number of channels that are increased with the width in practice (and therefore assumed to grow to infinity), the kernel size (k^2) can have an impact in practice, and this is the reason why it is included in the scalings.

A.8. Additional experiments

A.8.1. Nonlinear networks

Here we report some experiments with nonlinear networks trained on toy tasks, supporting our theory of linear networks.

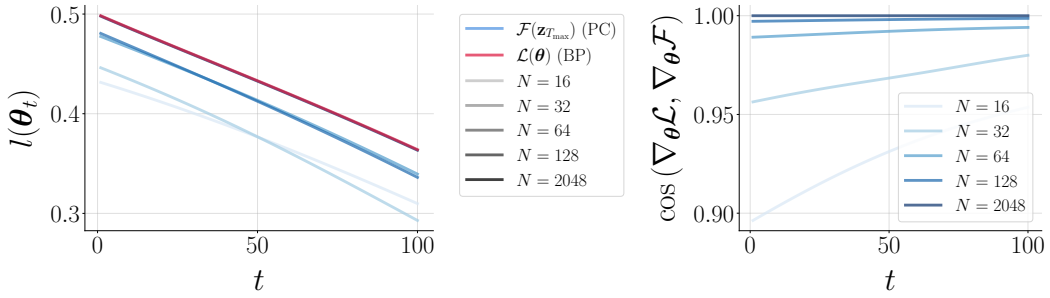


Figure A.1. **Under stable and feature-learning parameterisations, PC converges to BP for model width much larger than depth even on *nonlinear* networks.** With a similar setup to Figure 2, we plot results for nonlinear residual networks ($L = 5$) with Tanh as activation function. As in the linear case, we see that the PC energy at numerical convergence of the activities $\mathcal{F}(\boldsymbol{z}_{T_{\max}})$ converges to the BP MSE loss (Eq. 1) for sufficiently large width (*Left*), resulting in PC computing the same gradients as BP (*Right*). See the next Figure for results with ReLU and Figures A.21-A.22 for similar results with MLPs.

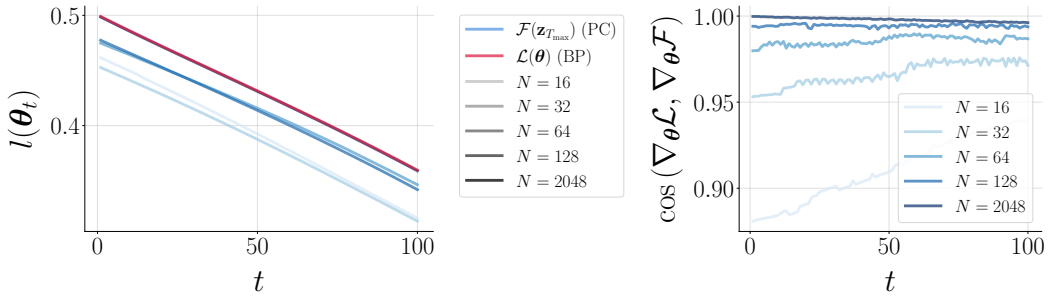


Figure A.2. **Results of Figure A.1 for ReLU.**

A.8.2. Learning regimes

Here we present supplementary results related to §3.2. For both linear MLPs and residual networks, Figures A.3 & A.23 empirically verify that, at large width and under stable parameterisations, PC shows similar learning regimes to BP (rich and lazy). Note that while the rich regime ($\gamma_0 \geq 1$) is exactly the same for PC and BP, the lazy regime of PC starts diverging from BP's as $\gamma \to 0$, since the rescaling $s(\boldsymbol{\theta})$ grows to infinity and the equilibrated energy converges to zero.

On the other hand, Figures A.4 & A.24 show that, in the regime of larger depth than width, the fast PC saddle-to-saddle regime studied in Innocenti et al. (2024a) exists for MLPs but not residual networks, respectively.

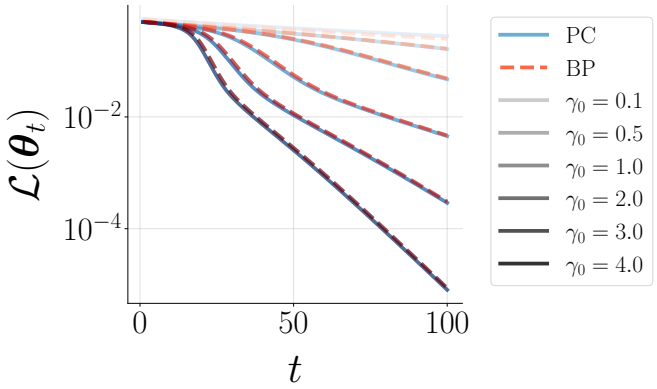


Figure A.3. **Stable feature-learning parameterisations of wide linear PCNs exhibit the same learning regimes (lazy vs rich) as BP at large width.** For the same setup of Figure 2, we fix the width at $N = 2048$ and compare the MSE loss (Eq. 1) of networks trained with both BP and PC for different values of the feature learning parameter γ_0 (Eq. 11). See Figure A.23 for similar results with residual networks.

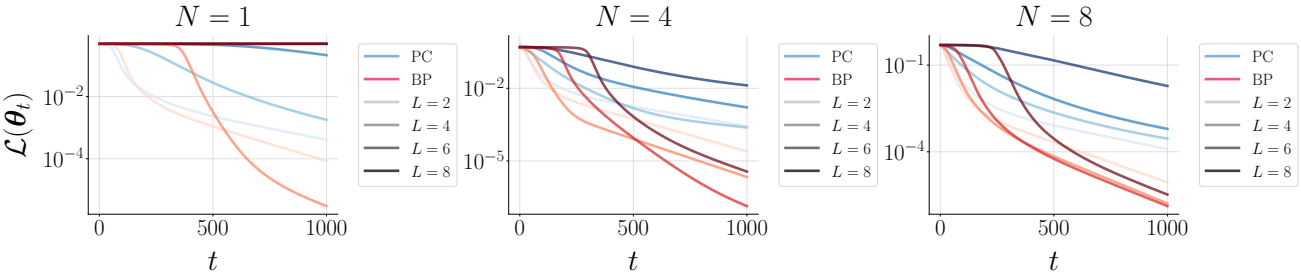


Figure A.4. **Under the SP, linear MLPs trained with PC have faster saddle escape dynamics for deeper than wide networks, $L > N$.** For the same setup of Figure 2, we plot the MSE loss (Eq. 1) for linear MLPs trained with BP and PC, varying both the width N and depth L. We see that PC shows faster escape times of the origin saddle for $L > N$. As discussed in §3.2, this regime does not exist for linear residual networks, as shown in Figure A.24.

A.9. Experimental details

Code to reproduce all the results is available at https://github.com/thebuckleylab/jpc/tree/main/experiments/limits_paper as part of a JAX library for training PCNs (Innocenti et al., 2024b). We always used no biases. For experiments with the SP, all networks used Kaiming Uniform $W_{ij}^{(\ell)} \sim \mathcal{U}(-1/\sqrt{d_{\text{in}}}, 1/\sqrt{d_{\text{in}}})$ as standard initialisation, with d_{in} as the input dimension. Wherever we do not report error bars, results were consistent across different random seeds.

Toy tasks. We trained linear networks with BP and PC to predict binary labels $y \in \{-1, 1\}$ with 40-dimensional inputs $\boldsymbol{x} \in \mathbb{R}^{40}$ drawn from $\mathcal{N}(0, 1)$. Unless otherwise stated, all experiments on this task used full-batch GD with $P = 20$ samples, learning rate constant $\eta_0 = 0.025$ (Eq. 14), and feature learning parameter $\gamma_0 = 1$ (Eq. 11).

For Figure 2, we trained MLPs of depth $L = 5$ and varying widths $N \in \{2^i\}_{i=3}^{11}$. The plots show selected widths to aid visualisation. For both PC and BP, we ran experiments with the mean-field parameterisation (Figures 2 & A.18) and the SP (Figures A.19-A.20), as defined in Table 2. Note that for the mean-field parameterisation the output scaling is $\gamma = \gamma_0\sqrt{N}$, and the learning rate is $\eta = \eta_0\gamma^2 N^{-c} = \eta_0\gamma_0^2 N$. For the experiments with residual networks (Figure A.17), we used $\alpha = 1/2$.

The theoretical infinite-width prediction and loss in Figure 2 were computed using the dynamical mean field theory developed by Bordelon & Pehlevan (2022b) (see in particular Appendix F). Under some approximations, this theory essentially allows one to integrate out the randomness over the weight initialisation and obtain a much lower-dimensional description of the network dynamics. For PC, we directly trained on (i.e. computed gradients of) the theoretical equilibrated energy (Eq. 5). Results with standard numerical optimisation of the activities (Eq. 3) are presented in §5 and detailed below.

For Figures A.3 & A.23, we fixed the width at $N = 2048$ and varied the feature learning parameter $\gamma_0 \in \{0.1, 0.5, 1, 2, 3, 4\}$ (Eq. 11). For Figures A.4 & A.24, we used the SP as in Innocenti et al. (2024a), and trained models of varying width and depth, with and without skips (respectively), for 1000 training iterations.

Image classification tasks. We trained networks to classify Fashion-MNIST and CIFAR-10. For the cosine similarity heatmaps (e.g. Figure 1) and rescaling plots (e.g. Figure 3), we trained mean-field-parameterised (see Table 2) linear and nonlinear networks, with and without skips, varying the width and depth. Cosine similarities were averaged over 3 random seeds. All experiments used Adam with batch size 64, and learning rate constant $\eta_0 = 1e^{-3}$.

Nonlinear models. For the results of Figure 4, we trained nonlinear residual networks, CNNs, and transformers with BP and PC. All experiments used 100 training steps, Adam and learning rate constant $\eta_0 = 1e^{-3}$. We plot cosine similarities during training for a representative run. The residual networks (MLPs) had 32 layers and Tanh as nonlinearity. For PC inference (Eq. 3), we used 100k iterations with step size $\beta = 0.3$ to ensure numerical convergence. All other training details for the residual MLPs, including the parameterisation (mean-field), were the same as in Figure 1.

The CNNs were trained on ImageNet with the cross-entropy loss and batch size 128. As explained in Figure 4, the width for these models is defined as the number of output channels times the kernel area, i.e. $N = k^2 C_{\text{out}}$. Following previous work (Bordelon et al., 2023), we kept the kernel size constant ($k = 3$) and varied the number of output channels $C_{\text{out}} \in \{2, 8, 16, 32\}$ of each convolutional layer. All the details of the architecture and parameterisation are given §A.7.1. For PC inference, we used 200 iterations with step size $\beta = 0.3$.

The transformers had a nano-GPT-style architecture with 12 blocks, 8 attention heads and varying width or embedding dimension N, and were trained on the character-level Tiny Shakespeare dataset with the cross-entropy loss. We used batch size 64 and a sequence length of 32. Details of the architecture and parameterisation are given §A.7.2. The only key difference with the standard nano-GPT model was the removal of layer normalisation (Ba et al., 2016), which we found to break the convergence of PC to BP even at large width, perhaps because of its interference with the PC inference dynamics. For visualisation purposes, we report results without weight decay, since we found that convergence to BP was even faster with the width when using weight decay. For PC inference, we used 800 iterations with step size $\beta = 0.45$.

A.10. Compute resources

The experiments with the widest and deepest models used a single NVIDIA RTX A6000 (48 GB VRAM). All other experiments were run on a CPU and took no more than a few hours, depending on the specific experiment.

A.11. Supplementary figures

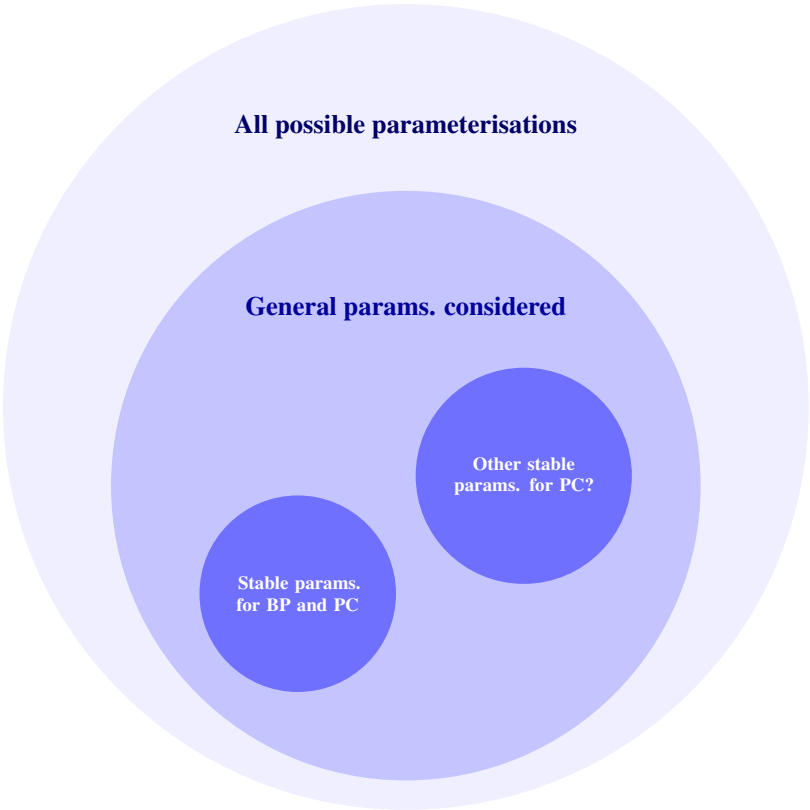


Figure A.5. **Venn diagram illustrating Theorems 1-2.** Following previous analyses for BP (Yang & Hu, 2021; Bordelon & Pehlevan, 2022b), we consider a specific set of general ("abcd") parameterisations of linear MLPs (Eqs. 7-10) and residual networks (Eq. 19), and show that the subset of stable (and feature-learning) parameterisations with respect to both model width and depth for PC is the same as for BP (Theorems 1-2). Within this subset, PC converges to BP for sufficiently large width (Corollaries 3.2 & 4.2). As discussed in §6, our results do not rule out other notions of stable and rich parameterisations (requiring different constraints), where PC might not converge to BP in some limit. Note also that, along with previous work, we do not consider all possible parameterisations.

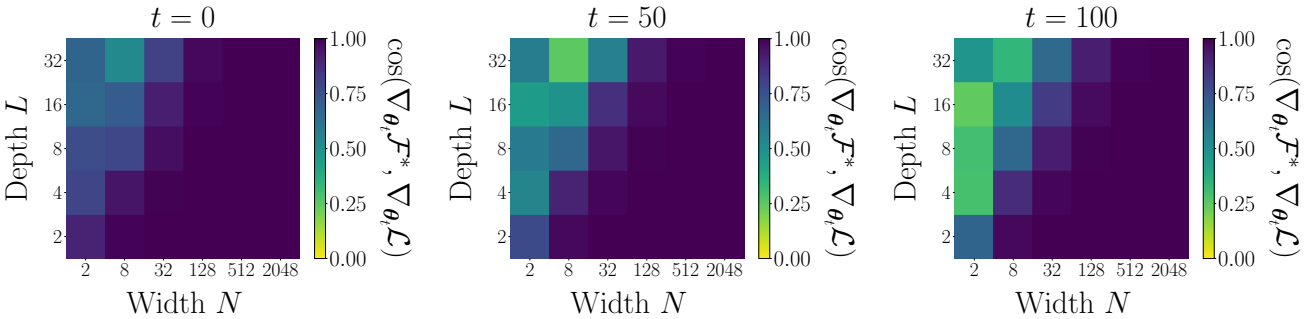


Figure A.6. **Results of Figure 1 for Fashion-MNIST.** A depth slice ($L = 32$) of these results is plotted in Figures A.11-A.12.

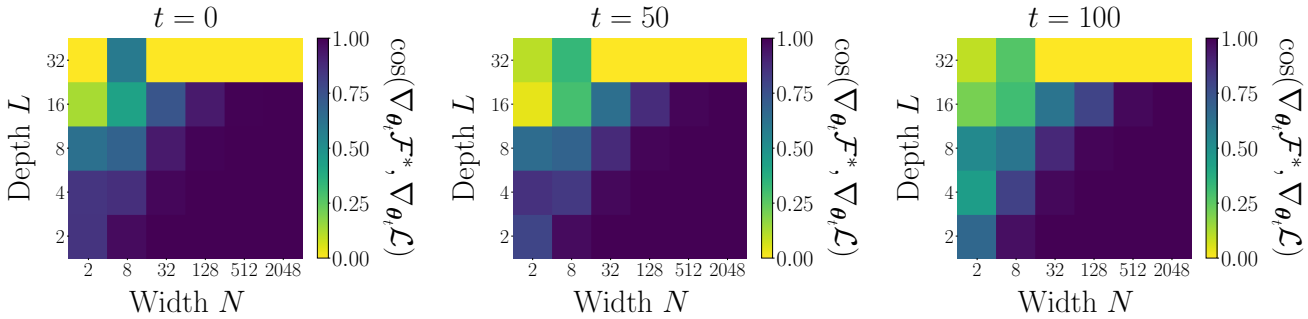


Figure A.7. **Results of Figure 1 for MLPs.** Unlike residual networks, the alignment between the PC and BP gradients breaks down at sufficiently large depth ($L = 32$) independent of the width, due the "fragility" of the MLP forward pass with depth as discussed in §3.2.

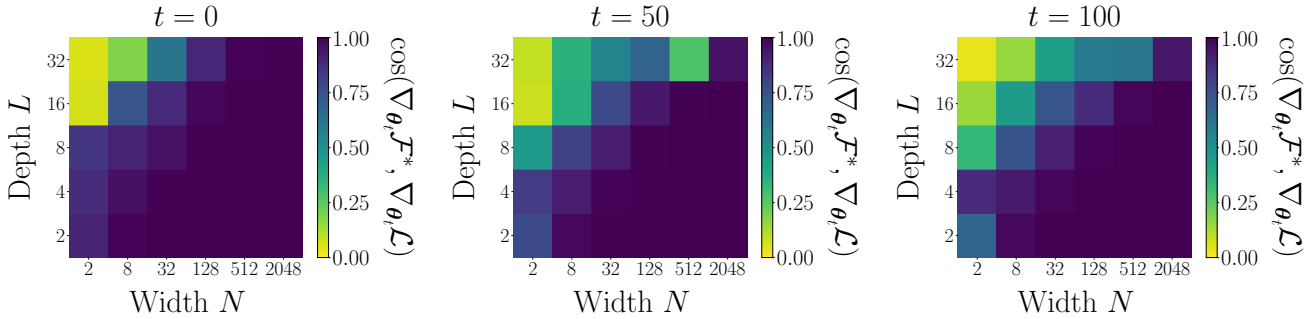


Figure A.8. **Results of the previous figure (A.7) for Fashion-MNIST.** Similar to the previous figure, we see that the PC gradients diverge from the BP gradients at large enough depth ($L = 32$), regardless of the width. Results for varying width at depth $L = 32$ are plotted in Figures A.13-A.14.

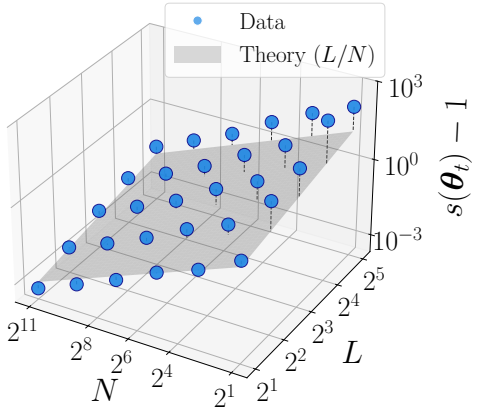


Figure A.9. **Results of Figure 3 for Fashion-MNIST.**

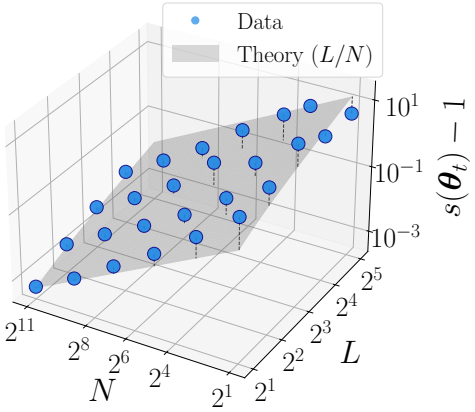


Figure A.10. **Results of Figure 3 for MLPs.**

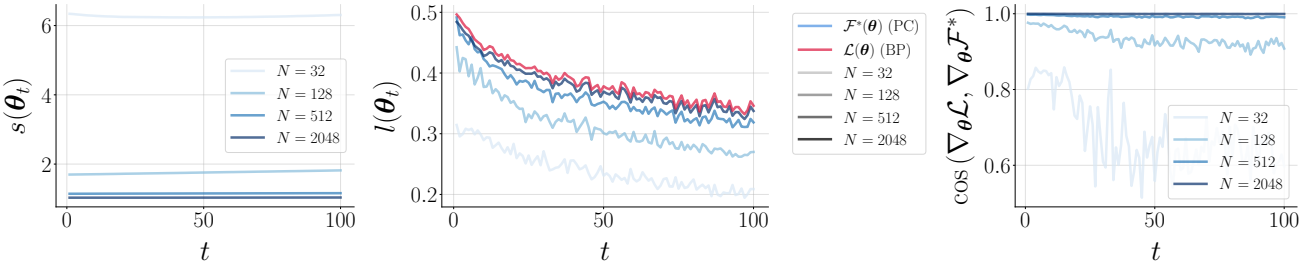


Figure A.11. **Depth slice ($L = 32$) results of Figure A.6.** See also the next figure.

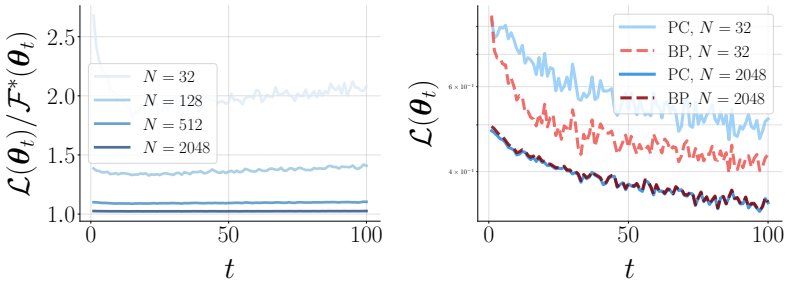


Figure A.12. **Additional results for Figure A.11.**

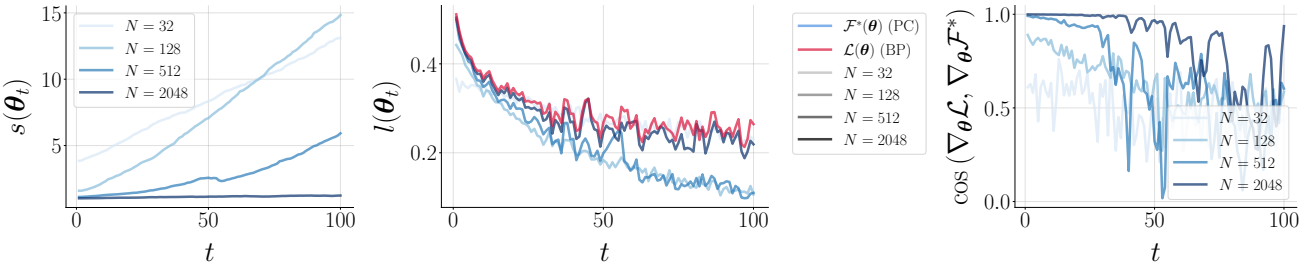


Figure A.13. **Results of Figure A.11 for MLPs.** Depth slice ($L = 32$) results of Figure A.8. As reflected in Figure A.8, PC starts to diverge from BP at sufficiently large depth, even at much larger width, because of the MLP forward pass instability with depth (§3.2). See also the next figure.

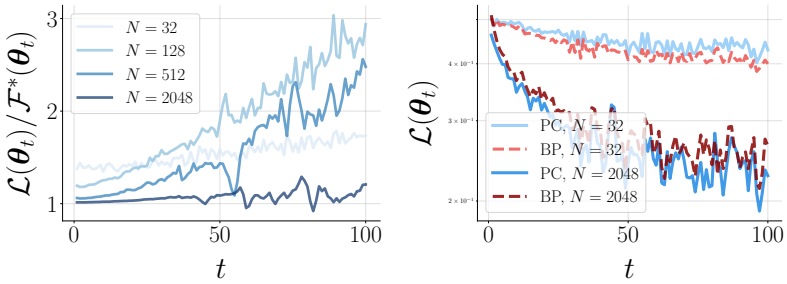


Figure A.14. **Additional results for Figure A.13.**

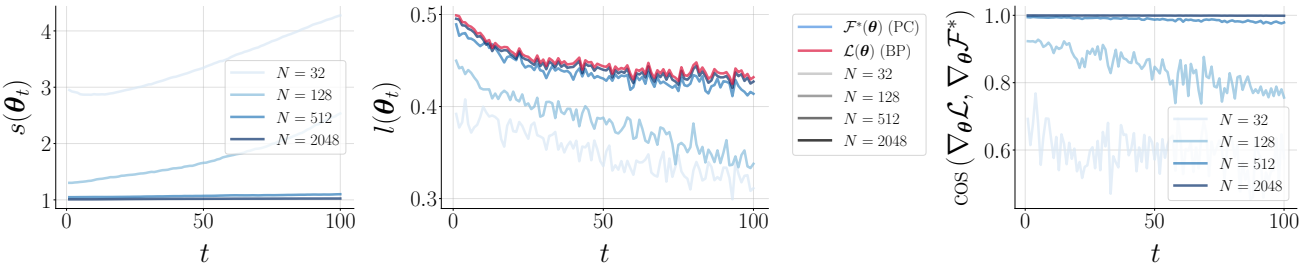


Figure A.15. **Similar results of Figure 2 on CIFAR-10.** We trained 16-layer MLPs of varying widths N on CIFAR-10 using SGD and the mean-field parameterisation (see Table 2). As in Figure 2, we observe: (i) that $s(\boldsymbol{\theta}) \to 1$ as $N \to \infty$ (*Left*); (ii) that the equilibrated energy (Eq. 5) converges to the MSE loss (Eq. 1) (*Middle*); and (iii) that the PC gradients converge to the BP gradients (*Right*). See §A.9 for more details.

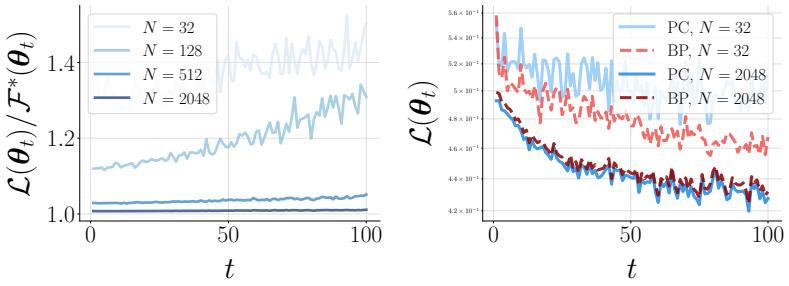


Figure A.16. **Additional results for Figure A.15.**

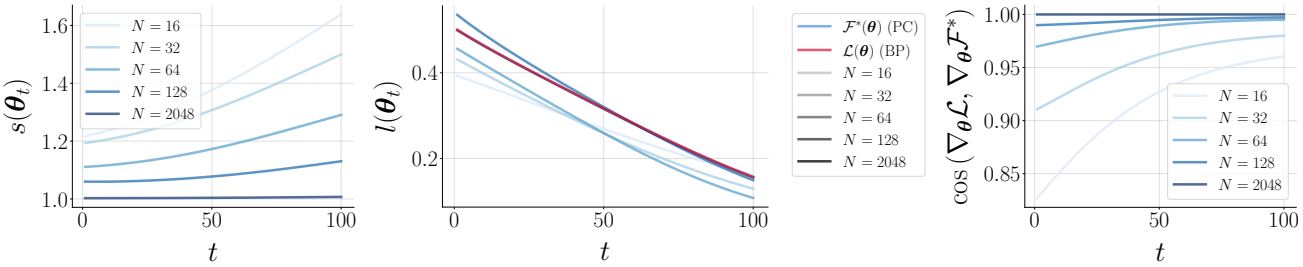


Figure A.17. **Results of Figure 2 for linear residual networks.**

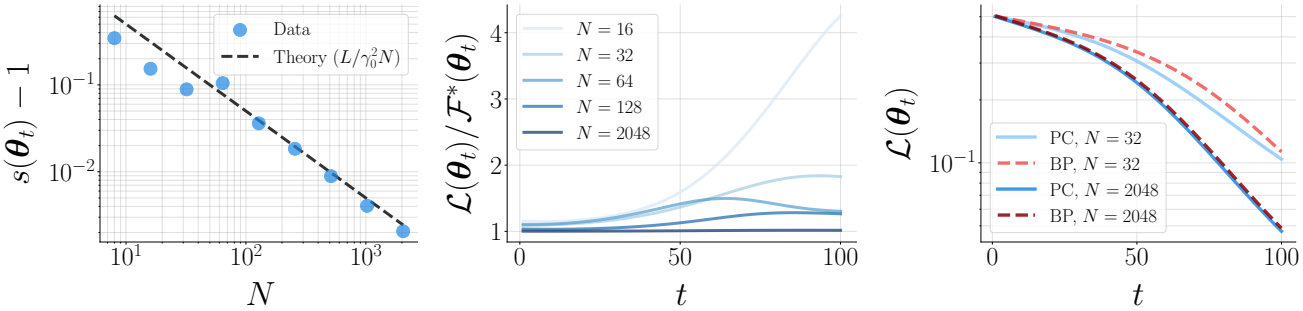


Figure A.18. **Additional results for Figure 2.** (*Left*) Comparison between the empirical and theoretical equilibrated energy rescaling $s(\boldsymbol{\theta})$ minus 1, verifying Eq. 17. Note that in this case the depth L and output scaling γ_0 (see Eq. 11) are constants with respect to the width and so do not affect the asymptotic scaling. (*Middle*) Ratio between the MSE loss (Eq. 1) and the equilibrated energy (Eq. 5) during training, for different network widths N. We see that the ratio converges to one as $N \to \infty$. (*Right*) MSE loss for a network trained with PC and BP for selected widths, showing convergence with large N.

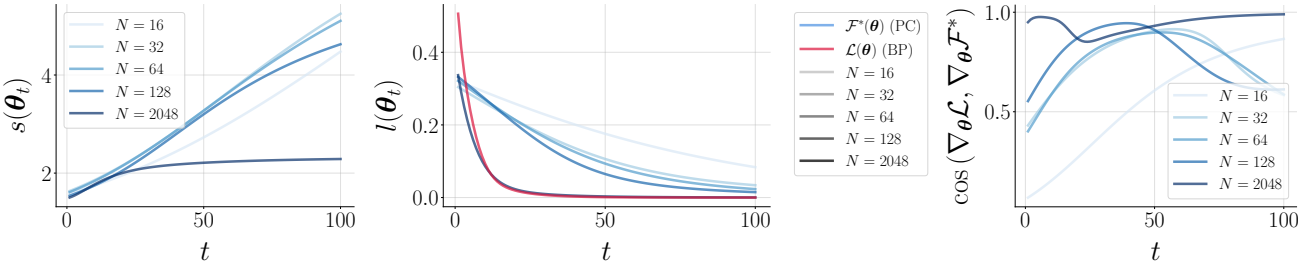


Figure A.19. **Results of Figure 2 for the SP.** In contrast to the results of Figure 2 with the mean-field parameterisation (see Table 2), (i) the rescaling $s(\boldsymbol{\theta})$ (Eq. 6) does not converge to one with the width (*Left*), (ii) the equilibrated energy (Eq. 5) does not converge to the MSE loss (Eq. 1) (*Middle*), and (iii) PC generally computes different gradients from BP (*Right*). Note that the gradients align for the widest networks because both the equilibrated energy and the loss are close to zero.

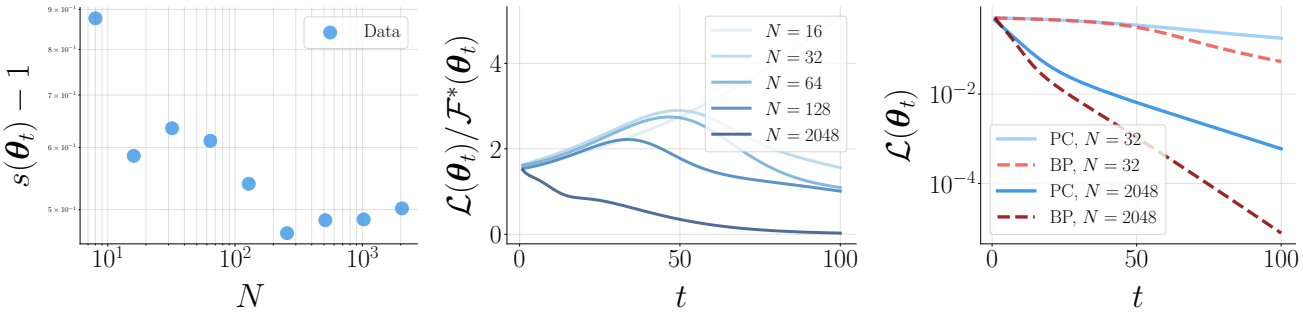


Figure A.20. **Results of Figure A.18 for the SP.** In contrast to comparable results with the mean-field parameterisation (Figure A.18), for the SP we observe (i) that the equilibrated energy rescaling $s(\boldsymbol{\theta})$ (Eq. 6) appears to be roughly constant with the width or $\Theta_N(1)$ (*Left*), and (ii) that there is no general correspondence between the equilibrated energy and the BP MSE loss at any width (*Middle* & *Right*).

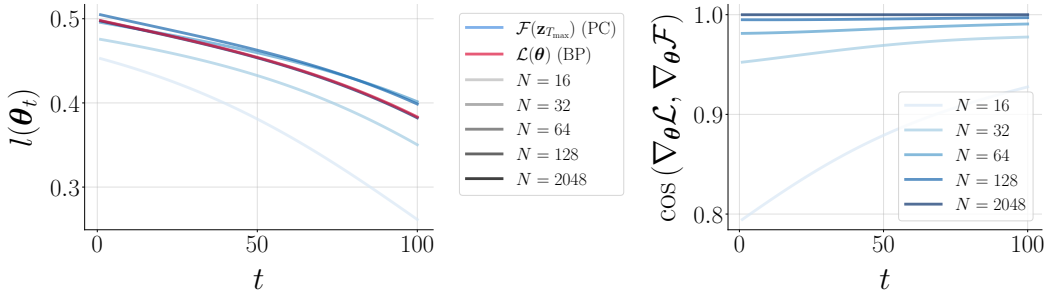


Figure A.21. **Results of Figure A.1 for Tanh MLPs.**

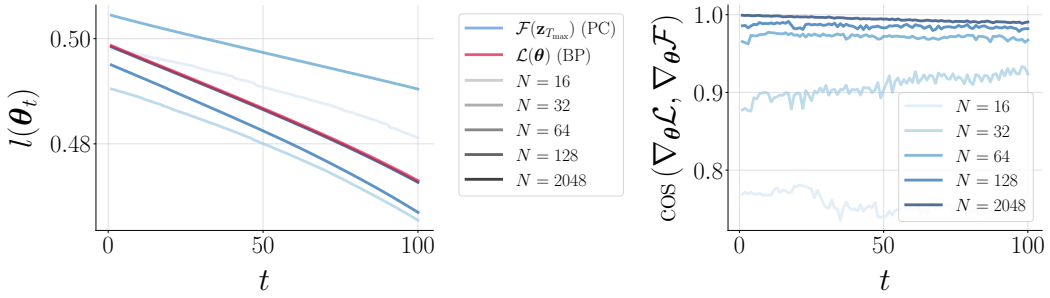


Figure A.22. **Results of Figure A.21 for ReLU.**

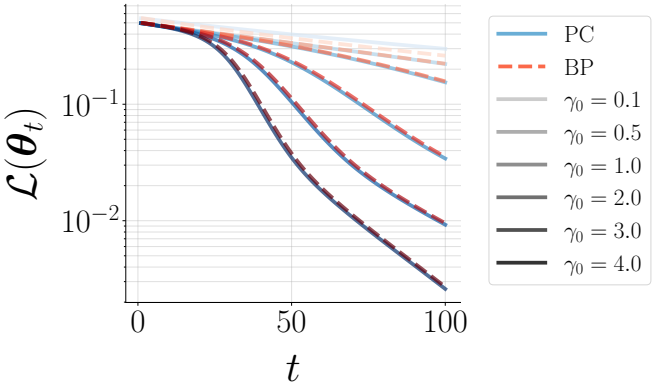


Figure A.23. **Results of Figure A.3 for linear residual networks.** As in Figure A.3, we observe that larger values of γ_0 (Eq. 11) are associated with faster (or richer) learning dynamics.

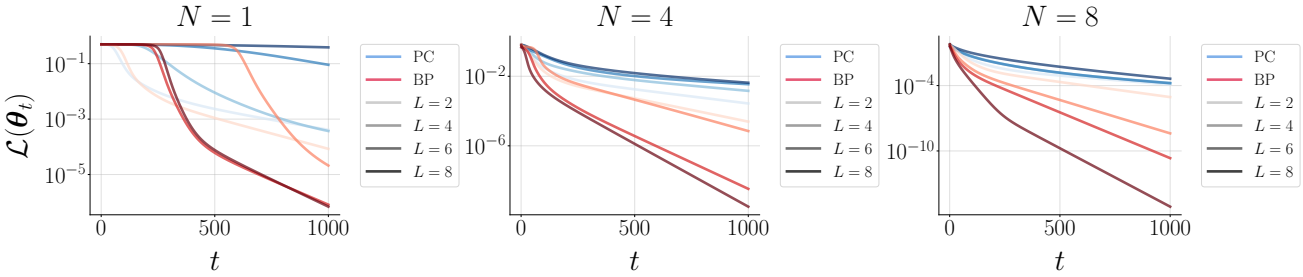


Figure A.24. **Results of Figure A.4 for linear residual networks.** As noted in §3.2, the fast PC saddle-to-saddle regime does not exist for residual networks (vs MLPs) since they effectively shift the position of the origin saddle.